Ref No. VTU/Aca/A6/2013-14/13107

Date: 7 MAR 2013

Circular

Sub: Revision of Scheme & Syllabus of PG Programme ‘Master of Computer Applications (MCA)’ for the year 2013-14
Ref: 1) Letter dated 14-6-2012 from the Chairman, BOS in MCA
2) Recommendations of the Joint Board of Studies Meeting held on 10-1-2013 and 11-1-2013 at VTU, Belgaum
3) Vice-Chancellor’s Order dated 6-3-2013

With reference to the above, this is to bring to your information that the Scheme & Syllabus of Master of Computer Applications (MCA) has been revised during the year 2010-11. As per the university policy, the revision of Scheme & Syllabus of MCA Programme has to be taken up for every 3 years. Accordingly the revision of Scheme & Syllabus of MCA Programme falls due from 2013-14.

In this connection, the Chairman, BOS in MCA has prepared the ‘Draft Scheme & Syllabus of MCA’ Course to be revised w.e.f. 2013-14 and submitted to the University. Now, the same has been uploaded in the University website under the head ‘Revision of Scheme & Syllabus of MCA Course (2013-14).

Hence, you are requested to bring this information to the concerned teaching faculty of your college for making comments / suggestions, if any on the ‘Draft Scheme & Syllabus’. Further, the same may be sent to the Chairman, BOS in MCA either in soft / hard copy at the following address, not later than 30-3-2013.

Dr. C.V. Srikrishna
Chairman, BOS in MCA of VTU, Belgaum &
Professor, P E S Institute of Technology
100ft Ring Road, 2nd Main
Banashankari, Hosakerehalli
Bangalore - 560 085
Phone: 080 – 26721983 Extn 228
Fax: 080 – 26720886
Cell: 9448107190
e-Mail: cvsrikrishna@yahoo.co.in

Also, one copy of comments / suggestions may be sent to the Registrar, VTU, Belgaum by post or e-Mail at registrar@vtu.ac.in

By order

Registrar

To
1. The Principals of all Engineering Colleges affiliated and the constituent Engineering College, VTU, Belgaum.
2. The PG Co-Ordinator, VTU Extension Centre / Regional Centre for PG Studies in Bangalore, Belgaum, Gulbarga & Mysore

Copy to:
2. The Special Officer, VTU’s Regional Office at Bangalore, Belgaum, Gulbarga & Mysore for information.
3. The Special Officer, Stores & Purchase Department, VTU, Belgaum for information.
4. In-Charge, CNC, VTU, Belgaum for uploading in the University website as informed in the Circular.
Scheme and Syllabus
(With effect from 2013-2014)

Master of Computer Applications

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM
## SCHEME OF TEACHING AND EXAMINATION
### MASTER OF COMPUTER APPLICATIONS

#### I SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Week</th>
<th>Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory Prac</td>
<td>IA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Duration Marks</td>
<td>Exam</td>
</tr>
<tr>
<td>1</td>
<td>13MCA11</td>
<td>Problem Solving using C</td>
<td>04 - 03</td>
<td>03 50 100 150</td>
</tr>
<tr>
<td>2</td>
<td>13MCA12</td>
<td>Discrete Mathematics &amp; Graph Theory</td>
<td>04 - 03</td>
<td>03 50 100 150</td>
</tr>
<tr>
<td>3</td>
<td>13MCA13</td>
<td>Digital Electronics and Computer Organization</td>
<td>04 - 03</td>
<td>03 50 100 150</td>
</tr>
<tr>
<td>4</td>
<td>13MCA14</td>
<td>Introduction to Unix</td>
<td>04 - 03</td>
<td>03 50 100 150</td>
</tr>
<tr>
<td>5</td>
<td>13MCA15</td>
<td>Introduction to Web Technologies</td>
<td>04 - 03</td>
<td>03 50 100 150</td>
</tr>
<tr>
<td>6</td>
<td>13MCA16</td>
<td>C Programming Laboratory</td>
<td>- 03</td>
<td>03 50 50 100</td>
</tr>
<tr>
<td>7</td>
<td>13MCA17</td>
<td>Unix Programming Laboratory</td>
<td>- 03</td>
<td>03 50 50 100</td>
</tr>
<tr>
<td>8</td>
<td>13MCA18</td>
<td>Web Programming Laboratory</td>
<td>- 03</td>
<td>03 50 50 100</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>20 09</td>
<td>400 650 1050</td>
</tr>
</tbody>
</table>
## SCHEME OF TEACHING AND EXAMINATION
### MASTER OF COMPUTER APPLICATIONS

#### II SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Week</th>
<th>Examination</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>Practical</td>
<td>Duration</td>
</tr>
<tr>
<td>1</td>
<td>13MCA21</td>
<td>Data Structures</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>2</td>
<td>13MCA22</td>
<td>Object Oriented Programming using C++</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>3</td>
<td>13MCA23</td>
<td>Operating Systems</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>4</td>
<td>13MCA24</td>
<td>Probability, Statistics and Queuing</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>5</td>
<td>13MCA25</td>
<td>Database Management Systems</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>6</td>
<td>13MCA26</td>
<td>Data Structures Using C Laboratory</td>
<td>-</td>
<td>03</td>
<td>03</td>
</tr>
<tr>
<td>7</td>
<td>13MCA27</td>
<td>Database Laboratory</td>
<td>-</td>
<td>03</td>
<td>03</td>
</tr>
<tr>
<td>8</td>
<td>13MCA28</td>
<td>OOP with C++ Laboratory</td>
<td>-</td>
<td>03</td>
<td>03</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>20</td>
<td>09</td>
<td></td>
</tr>
</tbody>
</table>
## III SEMESTER

**SCHEME OF TEACHING AND EXAMINATION**
**MASTER OF COMPUTER APPLICATIONS**

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Week</th>
<th>Examination Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>Practical</td>
</tr>
<tr>
<td>1</td>
<td>13MCA31</td>
<td>Computer Networks</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>13MCA32</td>
<td>Programming using Java</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>13MCA33</td>
<td>Software Engineering</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>13MCA34</td>
<td>Computer Graphics with Open GL</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>13MCA35</td>
<td>Elective-I</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>13MCA36</td>
<td>Java Programming Laboratory</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>7</td>
<td>13MCA37</td>
<td>CG Laboratory using Open GL</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>8</td>
<td>13MCA38</td>
<td>Network Laboratory</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td>20</td>
<td>09</td>
</tr>
</tbody>
</table>

### Elective I

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA351</td>
<td>UNIX system Programming</td>
</tr>
<tr>
<td>13MCA352</td>
<td>Advanced Topics in DBMS</td>
</tr>
<tr>
<td>13MCA353</td>
<td>Management Information Systems</td>
</tr>
<tr>
<td>13MCA354</td>
<td>Operations Research</td>
</tr>
<tr>
<td>13MCA355</td>
<td>Principles of User Interface Design</td>
</tr>
<tr>
<td>13MCA356</td>
<td>Systems Programming</td>
</tr>
</tbody>
</table>
## SCHEME OF TEACHING AND EXAMINATION
### MASTER OF COMPUTER APPLICATIONS

### IV SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Week</th>
<th>Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>Practical</td>
</tr>
<tr>
<td>1</td>
<td>13MCA41</td>
<td>Analysis and Design of Algorithms</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>13MCA42</td>
<td>Advanced Java Programming</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>13MCA43</td>
<td>Advanced Web Programming</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>13MCA44</td>
<td>Elective-II</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>13MCA45</td>
<td>Elective-III</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>10MCA46</td>
<td>ADA Laboratory</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>7</td>
<td>10MCA47</td>
<td>Advanced Java Programming Lab</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>8</td>
<td>10MCA48</td>
<td>Mini Project -I</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td><strong>20</strong></td>
<td><strong>09</strong></td>
</tr>
</tbody>
</table>

### Elective II

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA441</td>
<td>Advanced Computer Networks</td>
</tr>
<tr>
<td>13MCA442</td>
<td>Data Warehousing and Data Mining</td>
</tr>
<tr>
<td>13MCA443</td>
<td>Mobile Computing and Wireless</td>
</tr>
<tr>
<td></td>
<td>Communications</td>
</tr>
<tr>
<td>13MCA444</td>
<td>Software Testing and Practices</td>
</tr>
<tr>
<td>13MCA445</td>
<td>Theory of Computation (FAFL)</td>
</tr>
</tbody>
</table>

### Elective III

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA451</td>
<td>Cryptography &amp; Network Security</td>
</tr>
<tr>
<td>13MCA452</td>
<td>Network Management</td>
</tr>
<tr>
<td>13MCA453</td>
<td>NOSQL</td>
</tr>
<tr>
<td>13MCA454</td>
<td>Software Architectures</td>
</tr>
<tr>
<td>13MCA454</td>
<td>Enterprise Resource Planning (ERP)</td>
</tr>
</tbody>
</table>

---

**Note:** The marks distribution and teaching hours may vary depending on the institution's policy and curriculum requirements.
### SCHEME OF TEACHING AND EXAMINATION
### MASTER OF COMPUTER APPLICATIONS

#### V SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Week</th>
<th>Examination Marks</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>Practi</td>
<td>Duration</td>
</tr>
<tr>
<td>1</td>
<td>13MCA51</td>
<td>Object-Oriented Modeling and Design Patterns</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>2</td>
<td>13MCA52</td>
<td>System Simulation and Modeling</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>3</td>
<td>13MCA53</td>
<td>Programming using C#.NET</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>4</td>
<td>13MCA54</td>
<td>Elective II</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>5</td>
<td>13MCA55</td>
<td>Elective III</td>
<td>04</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>6</td>
<td>13MCA56</td>
<td>Software Design Laboratory</td>
<td>-</td>
<td>03</td>
<td>03</td>
</tr>
<tr>
<td>7</td>
<td>13MCA57</td>
<td>.Net Laboratory</td>
<td>-</td>
<td>03</td>
<td>03</td>
</tr>
<tr>
<td>8</td>
<td>13MCA58</td>
<td>Mini Project -II</td>
<td>-</td>
<td>03</td>
<td>03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>09</td>
<td>400</td>
</tr>
</tbody>
</table>

#### Elective IV
- 13MCA541 Mobile and Adhoc Sensor Networks
- 13MCA542 Parallel Computing
- 13MCA543 Multimedia systems
- 13MCA544 Pattern Recognition
- 13MCA545 Services Oriented Architecture
- 13MCA546 Compiler Design

#### Elective V
- 13MCA551 Cloud Computing
- 13MCA552 Web2.0 and Rich Internet Applications
- 13MCA553 Information Retrieval and Search Engines
- 13MCA554 Fuzzy Logic
- 13MCA555 Computer System Performance Analysis
- 13MCA556 Building Enterprise Applications
### VI SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Week</th>
<th>Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>Practical</td>
</tr>
<tr>
<td>1</td>
<td>13MCA61</td>
<td>Project Work</td>
<td>-</td>
<td>03</td>
</tr>
</tbody>
</table>

**NOTE:** Students have to register for one Elective from each of the Five Elective Groups (One from 3rd Semester, two from 4th Semester, 2 from 5th Semester)
Problem Solving Using C

**Unit-1**

12 hours

Algorithms, flowcharts, C structure, Identifiers, Types, Variables, Constants, Input/Output, Expressions, Precedence, Associativity, Side effects, Evaluating expressions, Type conversion, Statements.

Functions, User defined functions, Inter-function communication, Standard functions, Scope, Storage classes, Type qualifiers.

**Unit-2**

6 hours

Selection - logical data, operators, two way selection, Multiway selection, Repetition -Concept of a loop, pretest, post test loops, loops in C, examples, break, continue, applications of loops, Recursion

**Unit-3**

12 Hours

Files - streams, Standard library input/output functions, formatting input and output functions, character I/P2 O/P functions. Arrays - concepts, using arrays in C, inter-function communication, bubble sort, sequential, binary search, two dimensional array, multi dimensional arrays.

**Unit-4**

12 Hours

Pointers- Introduction, inter-function communication, pointers to pointers, compatibility, L value and R value, arrays, pointers, Pointer arithmetic and arrays, passing an array to a function memory allocations functions, array of pointers pointer to void, pointers to function. Strings-concepts, C strings, string I/P, O/P functions, arrays of strings, string Manipulation functions. String/Data conversion

**Unit-5**

10 Hours

Enumerated types - structure, union, Bitwise operators - Exact size integer types, logical bitwise operators, shift operators, masks, Files - Text vs binary files, standard library functions for files, Preprocessor commands - file inclusion, Macro definition, conditional compilation, other command line arguments.

**Text Books**

2. Programming in C – Reema Thareja, Oxford Higher Education

**Reference Books**

1. The C Programming language, Brian W Kernighan, Dennis M Ritchie, PHI, 2nd edition
Discrete Mathematics & Graph Theory

Subject Code: 13MCA12
Hours/Week : 04
Total Hours : 52
I.A. Marks : 50
Exam Hours: 03
Exam Marks: 100

Fundamentals of Logic 14 Hours
Basic Connectives and Truth Tables, Logic Equivalence- the laws of Logic, Logical Implications, Rules of Inference, The use of Quantifiers, Quantifier Definitions, Proofs of Theorems, Logic Puzzles and Analyzing Claim

Set Theory 6 Hours
Sets and Subsets Set Operations and the Laws of Set Theory, Counting and Venn Diagrams, Principles of Inclusion and Exclusion, Permutations and Combinations with repetition

Properties of Integers and Recurrence 7 Hours
Mathematical Induction, Recurrence Definition, Euclidean Algorithms, The first order Linear recurrence relation.

Relations and Functions 10 Hours
Cartesian products and Relations, Functions-Plain and One-to-One, Onto Functions, Stirling Numbers and the Second Kind, Special functions, The Pigeon-hole principle, Function composition and inverse functions.

Relations 5 Hours
Properties of Relations, Computer recognition-Zero One Matrices and Directed graphs, Posets and Hasse Diagrams

Graphs, Theory and Trees 10 Hours
Terminology, Definitions, Properties and Examples, Connectivity and Adjacency, Euler and Hamilton, Representation and Isomorphism, Planarity and Chromatic Number, Directed Graphs and Weighted Graphs, Rooted Trees, Trees and Sorting

Text Books
2. Eric Gosset “Discrete Mathematics with Proof” Wiley India, 2nd Edition
   (Chapter 1, Chapter 10:10.1-10.6)

Reference books
2. Y N Singh " Discrete Mathematical Structures" Wiley India, 1st ed, 2010
Digital Electronics and Computer Organization

Subject Code: 13MCA13  
I.A. Marks : 50
Hours/Week : 04  
Exam Hours: 03
Total Hours : 52  
Exam Marks: 100

Binary Systems  
6 Hours

Combinational Logic and Arithmetic Circuits  
12 Hours

Sequential Logic  
6 Hours
Introduction, Flip – Flops, Triggering of Flip- Flops, Registers, Shift Registers, Design of Counters.

Basic Structure of Computers  
7 Hours

Machine Instruction and Programs  
7 Hours
Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Examples from Assembly Language Programming.

Input/Output Organization  
6 Hours
Accessing I/O Devices, Interrupts, Processor Example, Buses. Case studies of any two latest micro processor and their operations

The Memory System  
6 Hours

Text Books:

Reference Books:


INTRODUCTION TO UNIX

Sub Code : 13MCA14  IA Marks : 50
Hrs/Week : 04  Exam Hours : 03
Total Hours : 52  Exam Marks: 100

Introduction of UNIX  9 Hours
Introduction, History, Architecture, Experience the Unix environment, Basic commands ls, cat, cal, date, calendar, who, printf, tty, sty, uname, passwd, echo, tcsh, script, spell and ispell, UNIX File System: The file, what’s in a filename? The parent-child relationship, pwd, the Home directory, absolute pathnames, using absolute pathnames for a command, cd, mkdir, rmdir, Relative pathnames, The UNIX file system.

Introduction to the Shell  7 Hours
Introduction to Shell Scripting, Shell Scripts, read, Command Line Arguments, Exit Status of a Command, The Logical Operators && and ||, exit, if, and case conditions, expr, sleep and wait, while, until, for, $, @, redirection. The here document, set, trap, Sample Validation and Data Entry Scripts.

Basic File Attributes  9 Hours
Is – l, the –d option, File Permissions, chmod, Security and File Permission, users and groups, security level, changing permission, user masks, changing ownership and group, File Attributes: process basics, PS, internal and external commands, running jobs in background, nice, at and batch, cron, time commands, More file attributes: hard link, symbolic link, umask, find

Simple Filters  8 Hours
Pr, head, tail, cut, paste, sort, uniq, tr commands, Filters using Regular Expression : grep & sed grep, Regular Expression, egrep, fgrep, sed instruction, Line Addressing, Inserting and Changing Text, Context addressing, writing selected lines to a file, the –f option, Substitution, Properties of Regular Expressions Context addressing, writing selected lines to a file, the –f option, Substitution, Properties of Regular Expressions

Awk-Advanced Filters  9 Hours
Simple awk Filtering, Splitting a Line into Fields, printf, the Logical and Relational Operators, Number Processing, Variables, The –f option, BEGIN and END positional Parameters, get line, Built-in variables, Arrays, Functions, Interface with the Shell, Control Flow, Advanced Shell Programming, The sh command, export, cd, the Command, expr, Conditional Parameter Substitution, Merging Streams, Shell Functions, eval, Exec Statement

The Process  8 Hours
Process basics, PS, internal and external commands, running jobs in background, nice, at and batch, cron, time commands, Essential System Administration root, administrator’s privileges,
Advanced System Administration 2 Hours

Case Study: emacs editor and any one distribution of Linux

Text Book:

1. Your UNIX-The Ultimate Guide, Sumitabha Das, Tata McGraw Hill,

Reference Book:

1. “Unix Shell Programming”, Yashwant Kanetkar,
2. “Beginning Shell Scripting”, Eric Foster -Johnson, John C Welch, Micah Anderson,
   Wrox publication.
Introduction to Web Technologies

Subject Code: 13MCA15                I.A. Marks: 50
Hours/Week: 4                           Exam Marks: 100
Total Hours: 52                          Exam Hours: 3

Fundamentals                          4 Hours
Internet, WWW, Web Browsers and Web Servers, URLs, MIME, HTTP, Security, the Web
Programmers Toolbox.

Web Foundations                       6 Hours
Evolution of the Web, Peak into the History of the Web, Internet Applications, Networks,
TCP/IP, Higher Level Protocols, Important Components of the Web, Web Search Engines,
Application Servers.

Introduction to XHTML                  10 Hours
Basic syntax, Standard structure, Basic text markup, Images, Hypertext Links. Lists, Tables,
Forms, Frames. Cascading Style Sheets: Introduction, Levels of style sheets, Style specification
formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment
of text, The box model, Background images, The <span> and <div> tags, Conflict resolution.

The Basics of JavaScript:              6 Hours
Overview of JavaScript, Object orientation and JavaScript, Syntactic characteristics, Primitives,
operations, and expressions, Screen output and keyboard input, Control statements, Object
creation and modification, Arrays, Functions, Constructors, Pattern matching using regular
expressions, Errors in scripts, Examples.

JavaScript and HTML Documents         6 Hours
Script, Events and Event Handling, Handling Events from Body Elements, Handling Events
from Text Box and password Elements, The DOM2 Event Model, The navigator Object, Dom
Tree Traversal and Modification.

Dynamic Documents with JavaScript:    6 Hours
Introduction, Positioning Elements, Moving Elements, Element Visibility, Changing Colors and
Fonts, Dynamic Content, Stacking Elements, Locating the Mouse Cursor, Reacting to a Mouse
Click, Slow Movement of Elements, Dragging and Dropping Elements.

Introduction to XML                   6 Hours
Introduction, Syntax, Document structure, Document type definitions, Namespaces, XML
schemas, displaying raw XML documents, Displaying XML documents with CSS, XSLT style
sheets, XML processors, Web services.
The Basics of Perl

Origins and uses of Perl, Scalars and their operations, Assignment statements and simple input and output, Control statements, Fundamentals of arrays, Hashes, References, Functions, Pattern matching, File input and output; Examples. Using Perl for CGI Programming: The Common Gateway Interface; CGI linkage; Query string format; CGI.pm module; A survey example; Cookies.

Text Books:

Reference Books:
Write a C Program to

1. a. Convert degrees into Fahrenheit’s and vice versa.
   b. Calculate the salary of an employee given his basic pay, HRA = 10% of basic pay, TA = 5% of his basic pay and deductions IT = 2.5% of his basic pay.

2. a. Check whether a number is a perfect number or not.
   b. Solve quadratic equations given the value of a, b and c.

3. a. Find Armstrong number.
   b. Convert a number to any base given.

4. Accept a string from user and encodes it. Apply the following procedure to encode the string.
   a. Convert each character in a string to its ASCI value.
   b. Add an integer value to it.
   c. Display the encoded string.
   d. Also decode the string into its original form using the reverse procedure and display the same.

5. Define a structure called student having properties like student id, student name and branch of student with a sub structure of marks of 3 subjects. Write a program which allows user to add new student, delete a student and also display all the students. Find the name of the students who have scored the best and worst marks. Also find the average marks scored by the students.

6. Take two integer n1 and n2 from user where n1<n2. Create functions that calculate the sum of all the integers ranging from n1 and n2, sum of all the odd numbers ranging from n1 and n2, sum of all the even numbers ranging from n1 and n2. Display an error message if n1>n2.

7. Create a structure Complex Number having real and imaginary part as properties. Write functions to add and subtract two complex numbers -.

8. a. Find a factorial of given number using recursion.
    b. Find the Fibonacci series using recursion.

9. Multiply two matrices that satisfy the constraint.
10. Find the saddle point in a matrix.

11. a. Find whether a matrix is identity or not.
   b. Find the transpose of a matrix.

12. a. Remove all the white spaces and newline character from a file.
   b. Find a given word in a file if it exists and also show the location of that word in a file.

13. Copy one file content to another file without using inbuilt functions.

14. Create a rainfall for a given string.

### UNIX Programming Laboratory

<table>
<thead>
<tr>
<th>Sub Code</th>
<th>13MCA17</th>
<th>L.A. Marks</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours/Week</td>
<td>3</td>
<td>Exam Hours</td>
<td>3</td>
</tr>
<tr>
<td>Total Hours</td>
<td>42</td>
<td>Exam Marks</td>
<td>50</td>
</tr>
</tbody>
</table>

A. Explore the unix environment.

B. Explore vi editor with vim tutor. Perform the following operations using vi editor, but not limited to:

1. insert character, delete character, replace character
2. Save the file and continue working
3. save the file a exit the editor
4. quit the editor
5. quit without saving the file
6. rename a file
7. insert lines, delete lines,
8. set line numbers
9. search for a pattern
10. move forward and backward

1a. Write a shell script that takes a valid directory name as an argument and recursively descend all the sub-directories, finds the maximum length of any file in that hierarchy and writes this maximum value to the standard output.

b. Write a shell script that accepts a path name and creates all the components in that path name as directories. For example, if the script is named mpc, then the command mpc a/b/c/d should create directories a, a/b, a/b/c, a/b/c/d.

2a. Write a shell script that accepts two file names as arguments, checks if the permissions for these files are identical and if the permissions are identical, output common permissions and otherwise output each file name followed by its permissions.

b. Write a shell script which accepts valid log-in names as arguments and prints their corresponding home directories, if no arguments are specified, print a suitable error message.
3a. Create a script file called `file-properties` that reads a file name entered and outputs its properties.

b. Write shell script to implement terminal locking (similar to the lock command). It should prompt the user for a password. After accepting the password entered by the user, it must prompt again for the matching password as confirmation and if match occurs, it must lock the keyword until a matching password is entered again by the user. Note that the script must be written to disregard BREAK, control-D. No time limit need be implemented for the lock duration.

4a. Write a shell script that accept one or more filenames as argument and convert all of them to uppercase, provided they exist in current directory.

b. Write a shell script that displays all the links to a file specified as the first argument to the script. The second argument, which is optional, can be used to specify in which the search is to begin. If this second argument is not present, the search is to begin in current working directory. In either case, the starting directory as well as all its subdirectories at all levels must be searched. The script need not include any error checking.

5a. Write a shell script that accepts as filename as argument and display its creation time if file exist and if it does not send output error message.

b. Write a shell script to display the calendar for current month with current date replaced by * or ** depending on whether the date has one digit or two digits.

6a. Write a shell script to find a file/s that matches a pattern given as command line argument in the home directory, display the contents of the file and copy the file into the directory `~/mydir`

b. Write a shell script to list all the files in a directory whose filename is at least 10 characters. (use expr command to check the length)

7a. Write a shell script that gets executed displays the message either “Good Morning” or “Good Afternoon” or “Good Evening” depending upon time at which the user logs in.

b. Write a shell script that accept a list of filenames as its argument, count and report occurrence of each word that is present in the first argument file on other argument files.

8a. Write a shell script that determine the period for which a specified user is working on system and display appropriate message.

b. Write a shell script that reports the logging in of a specified user within one minute after he/she log in. The script automatically terminate if specified user does not log in during a specified period of time.

9a. Write a shell script that accept the file name, starting and ending line number as an argument and display all the lines between the given line number.

b. Write a shell script that folds long lines into 40 columns. Thus any line that exceeds 40 characters must be broken after 40th, a “\" is to be appended as the indication of folding and the processing is to be continued with the residue. The input is to be supplied through a text file created by the user.

10a. Write an awk script that accepts date argument in the form of dd-mm-yy and displays it in the form of month, day and year. The script should check the validity of the argument and in the case of error, display a suitable message.
b. Write an awk script to delete duplicated line from a text file. The order of the original lines must remain unchanged.

11a. Write an awk script to find out total number of books sold in each discipline as well as total book sold using associate array down table as given below.

<table>
<thead>
<tr>
<th>Discipline</th>
<th>Books Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical</td>
<td>34</td>
</tr>
<tr>
<td>Mechanical</td>
<td>67</td>
</tr>
<tr>
<td>Electrical</td>
<td>80</td>
</tr>
<tr>
<td>Computer Science</td>
<td>43</td>
</tr>
<tr>
<td>Mechanical</td>
<td>65</td>
</tr>
<tr>
<td>Civil</td>
<td>98</td>
</tr>
<tr>
<td>Computer Science</td>
<td>64</td>
</tr>
</tbody>
</table>

b. Write an awk script to compute gross salary of an employee accordingly to rule given below.

If basic salary is < 10000 then HRA=15% of basic & DA=45% of basic

If basic salary is >=10000 then HRA=20% of basic & DA=50% of basic.

Note: In the examination each student picks one question from a lot of all the 11 questions. Question A & B Not to be included for examination

Web Programming Laboratory

<table>
<thead>
<tr>
<th>Subject Code: 13MCA18</th>
<th>I.A. Marks: 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours/Week: 3</td>
<td>Exam Hours: 3</td>
</tr>
<tr>
<td>Total Hours: 42</td>
<td>Exam Marks: 50</td>
</tr>
</tbody>
</table>

1. Create an XHTML page to demonstrate the usage of
   a. Text Formatting tags,
   b. Links
   c. Images
   d. Tables

2. Develop and demonstrate the usage of inline and external style sheet using CSS

3. Develop and demonstrate a XHTML file that includes JavaScript script for the following problems:
   a) Input: A number n obtained using prompt
      Output: The first n Fibonacci numbers
   b) Input: A number n obtained using prompt
      Output: A table of numbers from 1 to n and their squares using alert

4. Develop and demonstrate using JavaScript, a XHTML document that displays random numbers (integers).
5. a) Develop and demonstrate, using JavaScript script, a XHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two upper-case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.
   b) Modify the above program to get the current semester also (restricted to be a number from 1 to 8)

6. a) Develop and demonstrate, using JavaScript script, a XHTML document that contains three images, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible.
   b) Modify the above document so that when an image is moved from the top stacking position, it returns to its original position rather than to the bottom.

7. Develop using JavaScript script, an XHTML document that use of onload and onfocus events

8. a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Branch, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
   b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.

9. Write a Perl program which demonstrates the usage of scalar variables and arrays

10. Write a Perl program to display various Server information like Server Name, Server Software, Server protocol, CGI Revision etc.

11. Write a Perl program to display a digital clock which displays the current time of the server

12. Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.

13. Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

14. Write a CGI-Perl program to use a cookie to remember the day of the last login from a user and display it when run

**Note:** In the examination *each* student picks one question from the lot of *all 14* questions.
DATA STRUCTURES

Sub Code: 13MCA21
IA Marks: 50
Hrs/Week: 4
Exam Hours: 3
Total Hours: 52
Exam Marks: 100

Introduction to Data Structures
10 Hours
Information and its meaning: Abstract Data Types, Sequences as Value Definitions, ADT for Varying length character Strings, Data Types, Pointers and review of Pointers, Data Structures. Arrays: the Array as an ADT, Using One-dimensional Arrays, Implementing One-Dimensional Arrays, Arrays as Parameters, Handling of Character Strings and Character Strings.

The Stack
8 Hours
Definition and examples, Primitive operations, Example, The stack as an ADT, Representing stacks, Implementing the pop operation, Testing for exceptional conditions, Implementing the push operations, Examples for infix, postfix, and prefix expressions, Basic definition and Examples, Program to evaluate a postfix expression, Converting an expression from infix to postfix, Program to convert an expression from infix to postfix, Applications of Stacks: Regular Expressions, Expression Evaluations, Recursion etc.

Recursion
4 Hours
Recursive definition and processes, Factorial function, Multiplication of natural numbers, Fibonacci sequence, Binary search, Properties of recursive definition or algorithm. Binary search, Towers of Hanoi problem.

Queues and Lists
12 Hours
The queue and its sequential representation, the queue as ADT, Insert operation, Priority queue, Array implementation of a priority queue. Linked lists, Inserting and removing nodes from a list, Linked implementations of stacks, getnode and Freenode operations, Linked implementation of queues, Linked list as a data Structure, Example of list operations, Header nodes, Array implementation of lists, Limitations of array implementation, allocating and freeing dynamic variables, Linked lists using dynamic variables, Non integer and non-homogenous lists, Other list structures: Circular lists, Stack as a circular lists, doubly linked lists, Application of Linked Lists: Stacks, Queues, Double-ended Queues, Priority Queues.

Sorting
8 Hours
Bubble sort, Quick sort, Selection sort, Tree Sorting: Binary Tree Sort, Heap Sort, Insertion Sorts: Simple Insertion, Shell Sort, Address Calculation Sort, Merge and Radix Sort.

Searching
5 Hours
Basic Search Techniques: Algorithmic Notations, Sequential searching, Searching an ordered table, Indexed sequential search, Binary search, Interpolation search, Tree searching: Inserting into a Binary Search Tree, Deleting form a binary search tree, Hashing: Resolving hash clashes by open addressing, Choosing a hash Function.

Binary Trees
5 Hours
Tree traversals, Binary Search Tree and Operations, AVL Tree and Operations, Red-Black Tree, Threaded binary trees and operations.
Text Books:

Reference Books:

OBJECT ORIENTED PROGRAMMING USING C++

Sub Code: 13MCA22
IA Marks: 50
Hours/Week: 4
Exam Hours: 3
Total Hours: 52
Exam Marks: 100

Overview of OOP
Object Oriented paradigm, Structured vs. Object Oriented Paradigm. Elements of Object Oriented Programming, Object, Classes, Encapsulation & data abstraction, Inheritance, Polymorphism etc.

C++ Overview
Introduction, different data types, operators, expressions, qualifiers, arrays and strings, reference variables.

Modular Programming with Functions
Function Components, argument passing, inline functions, function overloading, function templates, class templates, recursive functions.

Classes & Objects
Introduction, Class Specification, Class Objects, access members, defining member functions, data hiding, constructors, destructors, parameterized constructors, static data members, functions, scope resolution operator, passing objects as arguments, returning objects, friend functions & classes, arrays of objects, Dynamic objects – Pointers to objects, Class members, Operator overloading using friend functions such as ++, --, [], etc.

Inheritance Virtual functions & Polymorphism & I/O Stream Library
Base Class, Inheritance & protected members, protected base class inheritance, inheriting multiple base classes, Constructors, Destructors & Inheritance. Passing parameters to base Class Constructors, Granting access, Virtual base classes, Virtual function -Calling a Virtual function through a base class reference, Virtual attribute is inherited, Virtual functions are hierarchical, pure virtual functions, abstract classes, using Virtual functions, Early & late binding. IO Stream Library, output operator <<, input >>, additional i/o operators, overloading the output operator <<, overloading the i/o operator >>, file input & output.

Exception Handling, STL
Exception handling fundamentals, Exception handling options, STL: An overview, containers, vectors, lists, maps.
Text Books:

Reference Book:

Operating Systems

Subject Code : 13MCA23 I.A. Marks : 50
Hours/Week : 4 Exam Hours : 3
Total Hours : 52 Exam Marks : 100

Introduction: Computer and Operating Systems 8 Hours

Operating System Structures: System Structures 6 Hours
System Components, Operating – System Services, System Calls, System Programs, System Structure, Virtual Machines, System Design and Implementation, System Generation.

Process Management 8 Hours

Mutual Execution and Synchronization 6 Hours
Principles of Concurrency, Mutual Exclusion: Hardware Support, Semaphores, Monitors, Message Passing, Readers/Writes Problem

Deadlock and Starvation 4 Hours
Principles of Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, An Integrated Deadlock Strategy, Dining Philosophers Problem

Memory Management 8 Hours
Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Process Creation, Page Replacement, Allocation of Frames, Thrashing

File – System Interface and Implementation 8 Hours

Secondary Storage, Computer Security 4 Hours

Note: Case Studies to be discussed for all Units.
Text Books


Reference Books


Probability, Statistics and Queuing

<table>
<thead>
<tr>
<th>Subject Code : 13MCA24</th>
<th>I.A. Marks   : 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours/Week : 4</td>
<td>Exam Hours : 3</td>
</tr>
<tr>
<td>Total Hours : 52</td>
<td>Exam Marks : 100</td>
</tr>
</tbody>
</table>

Introduction

Discrete Random Variables

Continuous Random Variables
Introduction, Random variables types, Functions of Random variables, Jointly distributed random variables, Functions of Normal Random variable, Probability mass function, Probability distribution function, Cumulative distribution function, Expected values of x, Moments, Moment generating function, Expectations based on Multiple Random variables, Discrete Distribution, Binomial distribution, Poison distribution, Geometric distribution, Continuous distribution, Normal Distribution, Exponential distribution, Weibull distribution, Distribution of Random variables, Joint Probability functions,

Stochastic Processes

Discrete-Time Markov Chains and Network of Queues
Statistical Inference                  5 Hours
Introduction, Parameter Estimation, Hypothesis Testing

Regression and Analysis of Variance                     7 Hours

Text Books

Reference Books
Chapter 1: Introduction to Database

Introduction to data, information, databases; types of databases; evolution of file system processing, its advantages and disadvantages; database systems, database system environment, dbms function, managing the database system.

Chapter 2: Data Modeling

Data modeling and data models, importance of data model, basics of data model, evolution of data models- hierarchical, network, relational, entity-relationship, object-oriented and xml; Data Abstraction- external model, conceptual model, physical model.

Chapter 3: Relational Database Model

Logical View of Data- tables and their characteristics, Keys- key, candidate key, primary key, foreign key, secondary key; Integrity Rules – entity integrity, referential integrity; Relational Set Operations- relational algebra -SELECT, PROJECT, JOIN, INTERSECT, UNION, DIFFERENCE, DIVIDE, PRODUCT; Data Dictionary and system catalog, relationship within relational database for unary, binary and ternary, composite entity(associative entity)

Chapter 4: Entity-Relationship Model

Introduction, entity, attributes, domain, identifiers, composite identifiers; Attributes: simple, composite, single valued attributes, composite attributes, multi-valued, derived and their implementation; Relationships- connectivity and cardinality, relationship strength- strong and weak entity, relationship participation, relationship degree, associative entities, developing an er-diagram

Chapter 5: Normalization of Database Tables

Database tables and normalization, need for normalization, normalization process-functional dependence- partial dependency, transitive depdendancy; FIRST Normal form- steps for follow in first normal law, SECOND Normal Form, THIRD Normal Form, Surrogate key consideration, BCNF, FOURTH normal form, Denormalization,
Chapter 6:
Structured Query Language (SQL)

Introduction to SQL, Data Definition commands, creating database, database schema, data types, creating table structures, SQL constraints, SQL indexes, Data Manipulation commands:- adding table rows, saving table changes, list table rows, updating, restoring table contents, deleting table rows, SELECT Queries, selecting rows with conditional constraints, arithmetic operators, logical operators, special operators, ordering a listing, functions- count, max, min, avg, sum, grouping data; Virtual tables- creating a view, joining database tables, recursive joins, outer joins, relational set operators- union, intersect, minus, SQL Join operators- cross join, natural join, join using clause, Subqueries and correlated queries- WHERE subqueries, IN subqueries, HAVING subqueries, multirow subquery operators: ANY and ALL, FROM subquery, ATTRIBUTE list subqueries, Correlated subquery, SQL Function- data and time function, numeric functions, string functions, conversion functions, introduction to triggers, stored procedure

Chapter 7
Transaction Management and Concurrency Control

Introduction to transaction, transaction properties, concurrency control; concurrency control with locking methods-lock granularity, lock types, two phase locking, deadlocks; concurrency control with time stamping methods- wait-die and wound-wait schemes; concurrency control with optimistic methods, database recovery management.-transaction recovery.

Text Book:
Database Principles Fundamentals of Design, Implementation and Management by Coronel, Morris, Rob - Chapter 1,2,3,4,5,6

Reference Book:
Write a C program to
1. Convert a prefix notation to postfix notation.
2. Evaluate a given postfix expression and its values for the variables.
3. Simulate the working of circular queue providing the following operations – Insert, Delete and Display.
4. Demonstrate recursion
   a. Calculate GCD and LCM of 3 integer numbers
   b. Solve Towers of Hanoi Problem
   c. Calculate the sum for a given number ‘n’ from 1 to n.
5. Simulate the working of a linked list providing the following operations
   a. Insert at the beginning
   b. Insert at the end
   c. Insert before a given element
   d. Insert at the position
   e. Display
6. Simulate the working of a circular linked list providing the following operations
   a. Delete from the beginning
   b. Delete from the end
   c. Delete a given element
   d. Delete every alternate element
   e. Display
   Insert is mandatory.
7. Simulate the working of a dequeue.
8. Simulate the working of a double linked list to implement stack and queue.
9. Create a binary tree and implement the tree traversal techniques of inorder, preorder and postorder.
10. Create a binary Tree and insert values into the tree.
11. Implement quick sort.
12. Implement Heap sort.
13. Implement the search techniques of
   a. Linear Search
   b. Binary Search
PART A

1. Write the ER design and Create the relational database of the Company with the below requirements and work out the queries
   Requirements (assume any required for the queries)
   - The company is organized into DEPARTMENTs. Each department has a name, number and an employee who manages the department. We keep track of the start date of the department manager.
   - Each department controls a number of PROJECTs. Each project has a name, number and is located at a single location.
   - We store each EMPLOYEE’s social security number, address, salary, sex, and birthdate. Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.
   - Each employee may have a number of DEPENDENTs. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Queries
Create all the relations based on the above scenario and do the following queries
   a. Retrieve the names of all employees who do not have supervisors
   b. Retrieve the names of all employees whose surname is same as their supervisors
   c. Retrieve the name of each employee who has a dependent with the same first name as the employee.
   d. Retrieve the name of each employee who works on all the projects controlled by department number 5.
   e. Retrieve the names of employees who have no dependents.
   f. For each project on which more than two employees work, retrieve the project number, project name, and the number of employees who work on that project.

2. Notown Records has decided to store information about musicians who perform on its albums (as well as other company data) in a database. Each musician that records at Notown has an SSN, a name, an address, and a phone number. Poorly paid musicians often share the same address, and no address has more than one phone.
   - Each instrument used in songs recorded at Notown has a unique identification number, a name (e.g., guitar, synthesizer, flute) and a musical key (e.g., C, B-flat, E-flat).
   - Each album recorded on the Notown label has a unique identification number, a title, a copyright date, a format (e.g., CD or MC), and an album identifier. Each song recorded at Notown has a title and an author.
   - Each musician may play several instruments, and a given instrument may be played by several musicians.
   - Each album has a number of songs on it, but no song may appear on more than one album.
   - Each song is performed by one or more musicians, and a musician may perform a number of songs. Each album has exactly one musician who acts as its producer. A musician may produce several albums, of course.

Design and develop a database for the above scenario and do the following
Queries

.a. List out the musician names, songs he has played, the album in which it has occurred and the title
.b. List out the albums which have the copyrights on the same day and has the same producer
.c. List out the different instruments played by the musicians and the average number of musicians who play the specific instrument
.d. Find out the album done by the producer of the album and plays guitar as well as flute and has produced no of songs greater than the average songs present
.e. List out how many musicians stay in address flat 29, RR nagar, Blore and what type of instruments they play
.f. Find the musicians who can play all the instruments present

3. Consider the following information about a university database: Professors have a PROFID, a name, an age, a rank, and a research specialty. Projects have a project number, a sponsor name (e.g., UGC/AICTE/…), a starting date, an ending date, and a budget. Graduate students have an USN, a name, an age, and a degree program (e.g., MCA/ MPhil/BE/ME …). Each project is managed by one professor (known as the project’s principal investigator).

Each project is worked on by one or more professors (known as the project’s co-investigators). Professors can manage and/or work on multiple projects. Each project is worked on by one or more graduate students (known as the project’s research assistants).

When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one. Departments have a department number, a department name, and a main office. Departments have a professor (known as the director) who runs the department. Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.

Graduate students have one major department in which they are working on their degree. Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take. Design and develop a database which reflects the above scenario and do the queries below

Queries
Create all the relations based on the above scenario and do the following queries

.a. Retrieve the names of all professors who do not have an ongoing project of more than 1 lakhs
.b. Retrieve the names of all graduate students along with their senior graduate student and the professors under whom they work for
.c. List the professors and the sum of their total budgeted projects
.d. Retrieve the names of project assistants who have more than two professors as supervisors and one of the supervisor is the director
.e. List the names of professors who has a total worth of project greater than the average budget of projects sanctioned
.f. List out the professors who work in more than one department and the time they work for each department

4. The Motor Vehicle Branch administers driving tests and issues driver’s licenses. Any person who wants a driver’s license must first take a learner’s exam at any Motor Vehicle Branch in the province. If he/she fails the exam, he can take the exam again any time after a week of the failed exam date, at any branch. If he passes the exam, he is issued a license (type = learner’s) with a unique license number. A learner’s license may contain a single restriction on it. The person may
take his driver’s exam at any branch any time before the learner’s license expiry date (which is usually set at six months after the license issue date). If he passes the exam, the branch issues him a driver’s license. A driver’s license must also record if the driver has completed driver’s education, for insurance purposes.

Create a E-R diagram following these steps.

- Find out the entities in the spec.
- Find out the relationships among the entities.
- figure out attributes of the entities and (if any) of the relationships.
- figure out constraints between entities and relationships.
- check to see if you don’t miss anything in spec.

Write the relational schema of the above problem.

1. Indentify the superkeys, candidate key, primary key and foreign keys in the relation.
2. Explain the cardinality and participation between the entities in the problem
3. Create the relations,
4. List out all persons who have taken the driving test and are not successful
5. List out a driver has taken test in more than 1 branch
6. list out drivers who have passed in one attempt

5. Design an ER diagarm for the following  requirements:-

Consider a hospital:

- Patients are treated in a single ward by the doctors assigned to them. Usually each patient will be assigned a single doctor, but in rare cases they will have two.

- Healthcare assistants also attend to the patients, a number of these are associated with each ward.

- Initially the system will be concerned solely with drug treatment. Each patient is required to take a variety of drugs a certain number of times per day and for varying lengths of time.

- The system must record details concerning patient treatment and staff payment. Some staff are paid part time and doctors and care assistants work varying amounts of overtime at varying rates (subject to grade).

- The system will also need to track what treatments are required for which patients and when and it should be capable of calculating the cost of treatment per week for each patient (though it is currently unclear to what use this information will be put).

1. Write the schema relation ship for the above problem
2. Identify superkey, candidate keys, primary keys,Referential Integrity
3. Explain the cardinality and participation between entities in the problem
4. Create the relations
5. list out the patients examined by a doctor
6. list out healthcare assistants of a ward
7. list out the cost of treatment per week by a patient.
6. Implement Triggers and Cursors Concept using SQL.

PART B

For each of the above scenarios construct suitable front end for querying and displaying the results

Instructions:

1. The exercises are to be solved in any RBBMS environment
2. Suitable tuples have to be entered so that queries are executed correctly
3. Questions of three queries can be asked in lots and any other relevant query can be asked by the examiner
4. The results of the queries may be displayed directly
5. Front end may be created in the examination based on the examiner

OOP with C++ Laboratory

Subject Code: 13MCA28
Hours/Week: 3
Total Hours: 42

1. Given that an EMPLOYEE class contains the following members:
   Data Members: Employee_Number, Employee_Name, Basic, DA, IT, Net_Sal, Member
   Functions: to read data, to calculate Net_Sal and to print data members, Write a C++ program to read data on N employees and compute the Net_Sal of each employee (DA = 52% of Basic and Income Tax = 30% of the gross salary)

2. Define a STUDENT class with USN, Name, and Marks in 3 tests of a subject. Declare an array of 10 STUDENT objects. Using appropriate functions, find the average of the two better marks for each student. Print the USN, Name and the average marks of all the students.

3. Write a C++ program to create a class called COMPLEX and implement the following overloading functions ADD that return a complex number:
   (i) ADD (a, s2) – where ‘a’ is an integer (real part) and s2 is a complex number
   (ii) ADD (s1, s2) – where s1 and s2 are complex numbers

4. Write a C++ program for scalar multiplication of two vectors using operator overloading.

5. Write a C++ program to create a template function for Bubble Sort and demonstrate sorting of integers and doubles.

6. Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators ‘+’ and ‘--’:
   (i) s1 = s1 + element; where s1 is an object of the class STACK and element is an integer to be pushed on the top of the stack
   (ii) s1 = --s1 ; where s1 is an object of the class STACK. ‘--‘operator pops the element. Handle the STACK empty and full conditions. Also display the contents of the stack after each operation, by overloading the << operator.

7. Create a class called MATRIX using two-dimensional array of integers. Implement the following operations by overloading the operator ++ which checks the compatibility of two matrices to be added and subtracted. Perform the addition and subtraction by overloading the +
and – operators respectively. Display the results by overloading the operator <<. If (m1==m2) then m3 = m1+m2 and m4 = m1-m2 else display error.

8. Write a C++ program to create a class called OCTAL which has the characteristics of an octal number. Implement the following operations by writing an appropriate constructor and an overloaded operator +.

(i) OCTAL h = x; where x is an integer.
(ii) int y = h + k; where h is an OCTAL object and k is an integer

Display the OCTAL result by overloading the operator <<. Also display the values of h and y.

9. Write a C++ program to create a class template called QUEUE with member functions to add an element and to delete an element from the queue. Using the member functions, implement a queue of integers and double. Demonstrate the operations by displaying the contents of the queue after every operation.

10. Define a class SET with Data members: array of int, int variable to indicate number of elements in a SET object; and Member functions: to read element of a SET object, to print elements of a SET object, to find union of 2 objects of SET using operator overloading (S3=S1+S2), to find intersection of 2 objects of SET using operator overloading (S3 = S1*S2). S1, S2, S3 and S4 are objects of SET. Use this class in a main function to show the above operations.

11. Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students. Find the semester-wise average age for all UG and PG students separately.

12. Write a C++ program to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<.

(i) STRING s1 = “VTU”
(ii) STRING s2 = “BELGAUM”
(iii) STRING s3 = s1 + s2 (Use copy constructor)

13. Define a base class STACK1 which performs only push, pop, display operations. Override the above operations through a derive class STACK2 which takes care of STACK FULL & STACK EMPTY situations. Show how the objects of these classes use the above functions in a main function.

14. Create an abstract base class EMPLOYEE with data members: Name, EmpID and BasicSal and a pure virtual function Cal_Sal().Create two derived classes MANAGER (with data members: DA and HRA) and SALESMAN (with data members: DA, HRA and TA). Write appropriate constructors and member functions to initialize the data, read and write the data and to calculate the net salary. The main() function should create array of base class pointers/references to invoke overridden functions and hence to implement run-time polymorphism.

Note: In the examination each student picks one question from a lot of all the 14 questions.
Computer Networks

Subject Code: 13MCA31                   IA Marks: 50
Hours/Week: 4                               Exam Hours: 3
Total Hours: 52                             Exam Marks: 100

Chapter: 1                                    08 Hours

Computer Networks and the Internet: what is the internet? The Network Edge, the Network Core, Delay, Loss and Throughput in Packet Switched Network, Protocol layers and their Service Modes.

Chapter: 2                                     12 Hours

Application Layer: Principles of Network Applications, the Web and HTTP, Electronic Mail in the Internet, DNS, Peer to Peer Applications, Socket Programming with TCP.

Chapter: 3                                     10 Hours

Transport Layer: Introduction and Transport layer services, Multiplexing and Demultiplexing, UDP, Principles of reliable data transfer, TCP, Principles of Congestion Control.

Chapter: 4                                     12 Hours

The Network Layer: Introduction, virtual circuits and Datagram networks, router architecture, Internet Protocol and IPv4 addressing, Link State Routing and Distance Vector Routing algorithms, RIP, OSPF & BGP

Chapter: 5                                     10 Hours

The Link Layer & LAN: Link layer and services, Errors detection & correction techniques, Multiple Access Protocols, link layer addressing and ARP, Ethernet, link layer switches.

Text Books

   - Chapter 1 1.1, 1.2, 1.3, 1.4, 1.5 Chapter 2(full), Chapter 3 (Except 3.6.3 and 3.7), Chapter 4 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 Chapter 5 5.1, 5.2, 5.3, 5.4, 5.5, 5.6


Reference Books

Programming using JAVA

Subject Code: 13MCA32       I.A. Marks: 50
Hours/Week: 4        Exam Hours: 3
Total Hours: 52        Exam Marks: 100

Java Programming Fundamentals

Introducing Data Types and Operators
Java’s Primitive Types, Literals, A Closer Look at Variables, The Scope and Lifetime of Variables, operators, Shorthand Assignments, Type conversion in Assignments, Using Cast, Operator Precedence, Expressions.

Program Control Statements
Input characters from the Keyword, The if statement, Nested ifs, The if-else-if Ladder, The Switch Statement, Nested switch statements, The for Loop, The Enhanced for Loop, The While Loop, The dowhile Loop, Use break, Use continue, Nested Loops

Introducing Classes, Objects and Methods
Class Fundamentals, How Objects are Created, Reference Variables and Assignment, Methods, Returning from a Method, Returning Value, Using Parameters, Constructors, Parameterized Constructors, The new operator Revisited, Garbage Collection and Finalizers, The this Keyword.

More Data Types and Operators
Arrays, Multidimensional Arrays, Alternative Array Declaration Syntax, Assigning Array References, Using the Length Member, The For-Each Style for Loop, Strings, The Bitwise operators.

String Handling
String Fundamentals, The String Constructors, Three String-Related Language Features, The Length() Method, Obtaining the characters within a string, String comparison, using indexOf() and lastIndexOf(), Changing the case of characters within a string, StringBuffer and String Builder.

A Closer Look at Methods and Classes
Controlling Access to Class Members, Pass Objects to Methods, How Arguments are passed, Returning Objects, Method Overloading, Overloading Constructors, Recursion, Understanding Static, Introducing Nested and Inner Classes, Varargs: Variable-Length Arguments.

Inheritance
Inheritance Basics, Member Access and Inheritance, Constructors and Inheritance, Using super to Call superclass constructors, Using super to Access Superclass Members, Creating a Multilevel Hierarchy, When are Constructors Executed, Superclass References and Subclass Objects, Method Overriding, Overridden Methods support polymorphism, Why Overridden Methods, Using Abstract Classes, Using final, The Object Class.

Interfaces
Interface Fundamentals, Creating an Interface, Implementing an Interface, Using Interface References, Implementing Multiple Interfaces, Constants in Interfaces, Interfaces can be extended, Nested Interfaces, Final Thoughts on Interfaces.

Packages
Package Fundamentals, Packages and Member Access, Importing Packages, Static Import.
Exception Handling 8 Hours
The Exception Hierarchy, Exception Handling Fundamentals, The Consequences of an Uncaught Exception, Exceptions Enable you to handle errors gracefully, using Multiple catch clauses, Catching subclass Exceptions, try blocks can be nested, Throwing an Exception, A Closer look at Throwable, using finally, using throws, Java’s Built-in Exceptions, New Exception features added by JDK 7, Creating Exception Subclasses.

Multithreaded Programming
Multithreading fundamentals, The Thread Class and Runnable Interface, Creating Thread, Creating Multiple Threads, Determining When a Thread Ends, Thread Priorities, Synchronization, using Synchronization Methods, The Synchronized Statement, Thread Communication using notify(), wait() and notify All(), suspending, Resuming and stopping Threads.

Enumerations, Auto boxing and Annotations 7 Hours
Enumerations, Java Enumeration are class types, The Values () and Valueof () Methods, Constructors, methods, instance variables and enumerations, Auto boxing, Annotations (metadata)

Generics
Generics Fundamentals Bounded Types, Generic Methods, Generic Constructors, Some Generic Restrictions.

Applets
Applet basics, A complete Applet Skeleton, Applet Initialization and Termination, A key Aspect of an Applet Architecture, Requesting Repainting, using the status window, Passing parameters to Applets.

Exploring java.lang 7 Hours
Primitive Type Wrappers, The Math class, The System class, The Object Class, The Class Class, Thread- Related Classes and Runnable Interface.

Exploring java.util
The Locale Class, Working with Date and Time, The Scanner Class, The Random Class.

Exploring Collection Framework
Collections Overview, The Collection Interfaces, The collection Classes. The Arrays Class.

Swing Fundamentals 7 Hours
The origin and Design philosophy of swing, Components and containers, Layout managers, A first simple swing Example, Event Handling, Exploring Swing Controls-JLabel and ImageIcon, The Swing Buttons, Trees.

Networking with Java.net

Text Books:
1. Java Fundamentals, A comprehensive Introduction by Herbert Schildt, Dale Skrien. Tata McGraw Hill Edition 2013. (Chapters: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,17,18,22,23,24,25 & 26)

Reference Books:
Software Engineering

Subject Code: 13MCA33
Hours/Week: 4
Total Hours: 52
I.A. Marks: 50
Exam Hours: 3
Exam Marks: 100

1. Overview
Introduction: Professional Software Development, Attributes of good software, software engineering diversity, IEEE/ACM code of software engineering ethics, case studies

2. Software Process
Software Process models: waterfall, incremental development, reuses oriented, Process activities; Coping with change, The rational Unified process.

3. Agile Software Development
Agile methods, Plan-driven and agile Development, Extreme Programming, Agile project management, Scaling agile methods.

4. Requirements Engineering
Functional and non-functional requirements, The software requirements document, Requirements specification, Requirements engineering processes, Requirement elicitation and analysis, Requirements validation, Requirements management

5. System Modeling
Context models, Interaction models, Structural models, Behavioral models, Model-driven engineering

6. Architectural Design

7. Design and implementation
Object-oriented design using UML, Design patterns, Implementation issues, Open source development.

8. Component-based software engineering
Components and component model, CBSE process, Component composition

9. Distributed Software engineering
Distributed system issues, Client-server computing, Architectural patterns for distributed systems, Software as a service.
10. Planning a software Project
Process planning, Effort estimation, Project scheduling and staffing, Software configuration management plan, Quality plan, Risk Management, Project monitoring plan. 04 hours

11. Software Testing
Testing fundamentals, Black-box testing, White-box testing, Testing process 04 hours

Text Books:
   (Chapters-: 1, 2, 3, 4, 5, 6, 7,17,18)
   (Chapters-: 5,10.1,10.2,10.3,10.4)

Reference Books:
Computer Graphics with Open GL

Subject Code : 13MCA34          I.A. Marks : 50
Hours/Week : 4                 Exam Hours : 3
Total Hours : 52        Exam Marks : 100

12 Hours

Graphics Output Primitives and Attributes
Introduction to open GL, Coordinate reference frames, Specifying two dimensional world coordinate reference frame in Open GL, Open GL point functions, Open GL line functions, Line drawing algorithms, Circle generation algorithms, Ellipse generation algorithms, Fill area primitives, Polygon fill areas, OpenGL polygon fill area functions, General scan line polygon fill algorithm, Fill methods for areas with irregular boundaries, Open GL fill area attribute functions

12 Hours

Two – Dimensional and Three - Dimensional Geometric Transformations
Basic two dimensional geometric transformations, Matrix representations and homogeneous coordinates, Inverse transformations, Two dimensional composite transformations, Other two dimensional transformations, Three dimensional Translation, Rotation, Scaling, Other three dimensional transformations, Affine transformations, Open GL geometric transformation functions

10 Hours

Two Dimensional Viewing
The two dimensional viewing, Clipping window, Normalization and viewport transformations, Clipping algorithms, Two dimensional point clipping, Two dimensional line clipping algorithms, Polygon fill area clipping, Curve clipping, Text clipping

10 Hours

Three Dimensional Viewing
The three dimensional viewing concepts, Three dimensional viewing pipeline, Three dimensional viewing coordinate parameters, Transformation from world to viewing coordinates, Projection transformations, Orthogonal projections, Oblique parallel projections, Perspective projections, The viewport transformation and three dimensional screen coordinates

08 Hours

Curves and Computer Animation
Bezier spline curves, Raster methods for computer animation, Design of animation sequences, Traditional animation techniques, General computer animation functions

Text book:
Chapters and topics [2.9, 3.1, 3.2, 3.3, 3.4, 3.5, 3.9, 3.10, 3.14, 3.15, 3.16, 4.10, 4.13, 4.14, 5.1, 5.2, 5.3, 5.4, 5.5, 5.10, 5.11, 5.12, 5.14, 5.15, 5.17, 6.1, 6.2, 6.3, 6.5, 6.6, 6.7, 6.8, 6.9, 6.10, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.10, 13.1, 13.2, 13.3, 13.4]

Reference Books:
UNIX System Programming

Subject Code: 13MCA351  I.A. Marks : 50
Hours/Week : 4  Exam Hours: 03
Total Hours : 52  Exam Marks: 100

Introduction  6 Hours

UNIX Files  6 Hours

UNIX File APIs  7 Hours
General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs, General File Class, regfile Class for Regular Files, dirfile Class for Directory Files, FIFO File Class, Device File Class, Symbolic Link File Class, File Listing Program.

UNIX Processes  7 Hours

Process Control  7 Hours

Signals and Daemon Processes  7 Hours
Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Single-instance daemons; Daemon conventions; Client-Server Model.

Interprocess Communication  6 Hours
Introduction; Pipes, popen, pclose Functions; Coprocesses; FIFOs; XSI IPC; Message Queues; Semaphores

Network IPC: Sockets  6 Hours
Introduction; Socket Descriptors; Addressing; Connection establishment; Data transfer; Socket options; Out-of-band data; Nonblocking and asynchronous I/O.
Text Books:
   (Chapters 1, 5, 6, 7, 8, 9)
   (Chapters 7, 8, 9, 13, 15, 16)

ADVANCED TOPICS IN DBMS

Sub Code: 13MCA352
Hrs/Week :04
Total Hours:52
IA Marks: 50
Exam Hours: 03
Exam Marks:100

Overview of Storage and Indexing, Disks and Files
7 Hours

Data on external storage; File organizations and indexing; Index data structures; Comparison of file organizations; Indexes and performance tuning; Memory hierarchy; RAID; Disk space management; Buffer manager; Files of records; Page formats and record formats.

Tree Structured Indexing
7 Hours

Intuition for tree indexes; Indexed sequential access; B+trees, Search, Insert, Delete, Duplicates, B+trees in practice.

Hash-Based Indexing
6 Hours

Static hashing, Extendible hashing, Linear hashing, comparisons.

Overview of Query Evaluation, External Sorting
6 Hours

The system catalog, Introduction to operator evaluation; Algorithm for relational operations; Introduction to query optimization; Alternative plans; A motivating example; what a typical optimizer does. When does a DBMS sort data? A simple two-way merge sort; External merge sort.

Evaluating Relational Operators
6 Hours

The Selection operation; General selection conditions; The Projection operation; The Join operation; The Set operations; Aggregate operations; The impact of buffering.

A Typical Relational Query Optimizer
7 Hours

Translating SQL queries in to Relational Algebra; Estimating the cost of a plan; Relational algebra equivalences; Enumeration of alternative plans; Nested sub-queries; other approaches to query optimization.

Physical Database Design and Tuning
7 Hours

Introduction; Guidelines for index selection, examples; Clustering and indexing; Indexes that enable index-only plans; Tools to assist in index selection; Overview of database tuning; Choices.
in tuning the conceptual schema; Choices in tuning queries and views; Impact of concurrency; DBMS benchmarking.

**More Recent Applications**

6 Hours

Mobile databases; Multimedia databases; Geographical Information Systems; Genome data management.

**Text Books:**


**Reference Books:**


---

**Management Information Systems**

<table>
<thead>
<tr>
<th>Subject Code: 13MCA353</th>
<th>I.A. Marks: 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours/Week: 4</td>
<td>Exam Hours: 3</td>
</tr>
<tr>
<td>Total Hours: 52</td>
<td>Exam Marks: 100</td>
</tr>
</tbody>
</table>

**Systems Engineering**

4 Hours

System concepts, system control, types of systems, handling system complexity, Classes of systems, General model of MIS, Need for system analysis, System analysis for existing system & new requirement, system development model, MIS & system analysis

**Information and Knowledge**

4 Hours

Information concepts, classification of information, methods of data and information collection, value of information, information: A quality product, General model of a human as information processor, Knowledge,

**Introduction of MIS**

4 Hours

MIS: Concept, Definition, Role of the MIS, Impact of MIS, MIS and the user, Management as a control system, MIS support to the management, Management effectiveness and MIS, Organization as system. MIS: organization effectiveness

**Strategic Management of Business**

3 Hours

Concept of corporate planning, Essentiality of strategic planning, Development of the business strategies, Type of strategies, short-range planning, tools of planning, MIS: strategic business planning

**Development of MIS**

4 Hours

Development of long range plans of the MIS, Ascertaining the class of information, Determining the information requirement, Development and implementation of the MIS, Management of information quality in the MIS, Organization for development of MIS, MIS development process model
Developing Business/IT Strategies/IT Solutions 5 Hours
Planning fundamentals (real world cases), Organizational planning, planning for competitive advantage, (SWOT Analysis), Business models and planning, Business/IT planning, identifying business/IT strategies, Implementation Challenges, Change management, Developing business systems, (real world case), SDLC, prototyping, System development process, implementing business system

Business Process Re-Engineering 2 Hours
Introduction, Business process, process model of the organization, value stream model of the organization, what delay the business process, relevance of information technology, MIS and BPR

Technology of Information System 4 Hours
Introduction, Data processing, Transaction processing, Application processing, information system processing, TQM of information systems, Human factors & user interface, Strategic nature of IT decision, MIS choice of information technology

Decision Making and DSS 3 Hours
Decision making concepts; decision making process, decision-making by analytical modeling, Behavioral concepts in decision making, organizational decision-making, Decision structure, DSS components, Management reporting alternatives.

Data resource Management 3 Hours
Managing data sources, Foundation concepts of data, types of databases, traditional file processing, DBMS approach, Database structure, Database development

Electronic Business systems 4 Hours
Enterprise business system – Introduction, cross-functional enterprise applications, real world case, Functional business system, - Introduction, marketing systems, sales force automation, CIM, HRM, online accounting system, Customer relationship management, ERP, Supply chain management (real world cases for the above)

Enterprise Business Systems 6 Hours
Electronic commerce fundamentals, e-Commerce applications and Issues, (real world cases)

Client Sever Architecture and E-business Technology 6 Hours

Text Books:

Reference Books:

Operations Research

Subject Code: 13MCA354                I.A. Marks: 50
Hours/Week: 4                        Exam Hours: 3
Total Hours: 52                      Exam Marks: 100

Introduction and Overview of the OR Modeling Approach
3 Hours
The origin of OR, the nature of OR, the impact of OR, defining the problem and gathering data, Formulating a mathematical model, deriving solutions from the model, testing the model, preparing to apply the model, implementation .

Introduction to Linear Programming
6 Hours
Formulation of linear programming problem (LPP), examples, Graphical solution, the LP Model, Special cases of Graphical method, assumptions of Linear Programming (LP), additional example

Solving LPP - the Simplex Method
12 Hours
The essence of the simplex method, setting up the simplex method, algebra of the simplex method, the simplex method in tabular form, special cases in the simplex method, tie breaking in the simplex method, adopting to other model forms (Two Phase method, Big-M method), post optimality analysis.

Theory of the Simplex Method
4 Hours
Foundation of the simplex method, the revised simplex method, a fundamental insight

Duality Theory and Sensitivity Analysis
9 Hours
The essence of duality theory, economic interpretation of duality, primal dual relationship, adapting to other primal forms, the role of duality in sensitive analysis, the dual simplex method

Transportation and Assignment Problems
6 Hours
The transportation problem, a stream line simplex method for the transportation problem, the assignment problem, a special algorithm for the assignment problem

Metaheuristics
6 Hours
The nature of Metaheuristics, Tabu Search, Simulated Annealing, Generating Algorithms

Game Theory
6 Hours
The formulation of two persons, zero sum games, solving simple games- a prototype example, games with mixed strategies, graphical solution procedure, solving by linear programming, extensions

Text Books:
   (Chapters 1.1, 1.2, 1.3, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 3.2, 3.3, 3.4, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 5.1, 5.2, 5.3, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 7.1, 8.1, 8.2, 8.3, 8.4, 13.1, 13.2, 13.3, 13.4, 14.1, 14.2, 14.3, 14.4, 14.5, 14.6)
Reference Books:

Principles of User Interface Design

Subject Code: 13MCA355   I.A. Marks: 50
Hours/Week: 4       Exam Hours: 3
Total Hours: 52       Exam Marks: 100

Introduction                     8 Hours
Usability of Interactive Systems: Introduction, Usability Requirements, Usability Measures, Usability Motivations, Universal Usability, Goals for our profession
Guideline, principles, and theories: Introduction, Guidelines, principles, Theories, Object-Action Interface Model

Development Processes           5 Hours

Evaluating Interface Design     7 Hours
Introduction, Expert Reviews, Usability Testing and Laboratories, Survey Instruments, Acceptance tests, Evaluation during Active Use, Controlled Psychologically Oriented Experiments

Interaction Styles             8 Hours
Direct Manipulation and Virtual Environments: Introduction, Examples of Direct Manipulation, 3D Interfaces, Teleoperation, Virtual and Augmented Reality
Menu Selection, Form Fillin and Dialog Boxes: Introduction, Task-Related Menu Organization, Single Menus, Combination of Multiple Menus, Content Organization, Fast Movement Through Menus, Data Entry With Menus, Form Fillin, Dialog Boxes and Alternatives, Audio Menus and Menus for Small Displays

Command and Natural Languages  8 Hours
Introduction, Functionality to support User’s tasks, Command-organization strategies, The benefits of Structure, Naming and Abbreviations, Natural Language in computing.
Interaction Devices: Introduction, Keyboards and Keypads, Pointing Devices, Speech and Auditory interfaces, Displays-Small and Large, Printers

Design Issues                   6 Hours
Quality of Service: Introduction, Models of Response-Time Impacts, Expectations and Attitudes, User Productivity, Variability in Response time, Frustrating Experiences

User Manuals, Online Help and Tutorials     5 Hours
Introduction, Paper versus Online Manuals, Reading from paper versus Displays, Shaping the content of the Manuals, Online Manuals and Help, Online Tutorials, Demonstrations and Guides, Online Communities for User Assistance, The Development Process.

**Information Search and Visualization**
5 Hours
Introduction, Search in Textual Documents and Database Querying, Multimedia document searches, Advanced filtering and Search Interfaces, Information Visualization

**TextBooks**

**Reference Books**

**Systems Programming**

<table>
<thead>
<tr>
<th>Subject Code: 13MCA356</th>
<th>I.A. Marks : 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours/Week : 04</td>
<td>Exam Hours: 03</td>
</tr>
<tr>
<td>Total Hours : 52</td>
<td>Exam Marks: 100</td>
</tr>
</tbody>
</table>

**Machine Architecture**
6 Hours

**Assemblers**
12 Hours

**Loaders and Linkers**
8 Hours

**Editors And Debugging Systems**
6 Hours
Text Editors - Overview of Editing Process, User Interface, Editor Structure, Interactive Debugging Systems - Debugging Functions and Capabilities, Relationship With Other Parts Of The System, User-Interface Criteria

**Macro Processor**
8 Hours
Basic Macro Processor Functions - Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures, Machine-Independent Macro Processor Features - Concatenation


Text Books:
1. Leland.L.Beck: System Software, 3rd Edition, Addison-Wesley, 1997. (Chapters 1.1 to 1.3, 2 (except 2.5.2 and 2.5.3), 3 (except 3.5.2 and 3.5.3), 4 (except 4.4.3))

Reference Books:

Java Programming Laboratory

Subject Code: 13MCA36
Hours/Week: 3
Total Hours: 42
L.A. Marks: 50
Exam Hours: 3
Exam Marks: 50

1. a. Write a JAVA Program to demonstrate Constructor Overloading and Method Overloading.
b. Write a JAVA Program to implement Inner class and demonstrate its Access Protections.
2. Write a JAVA Program how to find convert an array of int to a String in Java.
3. Write a program in Java for String handling which performs the following:
   i) Checks the capacity of StringBuffer objects.
   ii) Reverses the contents of a string given on console and converts the resultant string in upper case.
   iii) Reads a string from console and appends it to the resultant string of ii.
4. a. Write a JAVA Program to demonstrate Inheritance.
b. Simple Program on Java for the implementation of Multiple inheritance using interfaces to calculate the area of a rectangle and triangle.
5. Write a JAVA program which has
   i. A Class called Account that creates account with 500Rs minimum balance, a deposit() method to deposit amount, a withdraw() method to withdraw amount and also throws LessBalanceException if an account holder tries to withdraw money which makes the balance become less than 500Rs.
   ii. A Class called LessBalanceException which returns the statement that says withdraw amount (___Rs) is not valid.
iii. A Class which creates 2 accounts, both account deposit money and one account tries to withdraw more money which generates a LessBalanceException take appropriate action for the same.

6. Write a JAVA program using Synchronized Threads, which demonstrates Producer Consumer concept.

7. Write a JAVA program to implement a Queue using user defined Exception Handling (also make use of throw, throws.).

8. Complete the following:
   1. Create a package named shape.
   2. Create some classes in the package representing some common shapes like Square, Triangle, and Circle.
   3. Import and compile these classes in other program.

9. a. Create an enumeration Day of Week with seven values SUNDAY through SATURDAY. Add a method is Workday( ) to the DayOfWeek class that returns true if the value on which it is called is MONDAY through FRIDAY. For example, the call DayOfWeek.SUNDAY.isWorkDay ( ) returns false.
   b. Write JAVA Applet programs which handles KeyboardEvent

10. Write a JAVA program which has
    i. A Interface class for Stack Operations
    ii. A Class that implements the Stack Interface and creates a fixed length Stack.
    iii. A Class that implements the Stack Interface and creates a Dynamic length Stack.
    iv. A Class that uses both the above Stacks through Interface reference and does the Stack operations that demonstrates the runtime binding.

11. Write a JAVA program to print a chessboard pattern.

12. Write a JAVA Program which uses FileInputStream / FileOutputStream Classes.

13. Write JAVA programs which demonstrates utilities of LinkedList Class.

14. Write a JAVA program which uses Datagram Socket for Client Server Communication.

15. Write a java program that lets user’s pie charts. Design your own user interface (with Swings & AWT).

---

**CG Laboratory using Open GL**

<table>
<thead>
<tr>
<th>Subject Code : 13MCA37</th>
<th>I.A. Marks : 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours/Week: 3</td>
<td>Exam Hours : 3</td>
</tr>
<tr>
<td>Total Hours: 42</td>
<td>Exam Marks : 50</td>
</tr>
</tbody>
</table>

**PART -A**

1. Write a program to create a chess board using DDA line algorithm
2. Write a program to implement Bresenham’s line drawing algorithm with all values of slopes
3. Write a program to implement Midpoint circle generation algorithm
4. Write a program to create a wireframe model of globe using equation of ellipse
5. Write a program to create and fill the two dimensional object by using boundary fill algorithm
6. Write a program to create (with out using built in function) a cube by implementing translation algorithm by translating along 1. X-axis, 2.Y-axis and 3. X and Y plane
7. Write a program to create (with out using built in function) and rotate (1. given an angle 2. Around x and y axis) a triangle by implementing rotation algorithm
8. Write a program to create (with out using built in function) a triangle by implementing scaling algorithm by zooming/un-zooming along 1. X-axis, 2. Y-axis and 3. X and Y plane.

9. Write a program to create (with out using built in function) a Cube by implementing reflection algorithm 1. X-axis, 2. Y-axis.

10. Write a program to create (with out using built in function) a square by implementing shear algorithm along 1. X-axis, 2. Y-axis.

11. Write a program to animate a flag using Bezier Curve algorithm.

12. Write a program to clip lines using Liang-Barsky algorithm.

PART-B

1. Develop different chart options with the given inputs by applying DDA algorithm.
2. Develop different line styles using Bresenham’s algorithm.
3. Develop different circular patterns using midpoint circle generation algorithm.
4. Animate cube and globe with given attributes.
5. Develop a screen saver with curves with given attributes.
6. Develop a screen saver with text with given attributes.
7. Develop a screen saver with 2D objects.
8. Develop a screen saver with bouncing of 3D objects.
10. Animate a bicycle / car with given attributes.

Note:
Students Should Complete All Programs from Part-A and Any Two Programs from Part - B using Open GL. Consider all types attributes like color, thickness, styles, font, background, speed etc while doing Part - B.

In the examination each student picks one question from the lot of all 12 questions from Part – A and demonstrate any program from Part-B.

Online reference:

NETWORK LABORATORY

Subject Code: 13MCA38       IA Marks: 50
Hours/Week: 3                Exam Marks: 50
Total Hours: 42              Exam Hours: 03

1. Implementation of Hamming Code
2. Implementation of Leaky Bucket Algorithm
3. Socket Programming- TCP
4. Implementation Distance Vector Routing Algorithm
5. Implementation of ARP Protocol
6. Implementation of Sliding Window Protocol
7. Simulation of LAN
8. Simulation of Wireless LAN
9. Point-to-Point Link –NS2
10. Dynamic Routing Algorithm NS-2
11. Application of Socket Programming
12. Shortest Path Routing Algorithm

Simulator Experiments may be done on NS-2 or OPNET or NCTUNS or any other equivalent simulator or programming experiments are done using C++ or JAVA.

ANALYSIS AND DESIGN OF ALGORITHMS

<table>
<thead>
<tr>
<th>Sub Code : 13MCA41</th>
<th>IA Marks : 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/Week: 04</td>
<td>Exam Hours: 03</td>
</tr>
<tr>
<td>Total Hours: 52</td>
<td>Exam Marks: 100</td>
</tr>
</tbody>
</table>

**Introduction**  
Notion of Algorithm, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental data Structures.

**Fundamentals of the Analysis of Algorithm Efficiency**  
Analysis Framework, Asymptotic Notations and Basic efficiency classes, Mathematical analysis of Recursive and Nonrecursive algorithms, Examples.

**Brute Force**  
Selection Sort and Bubble Sort, Sequential Search and String Matching, Exhaustive Search.

**Divide-and-Conquer**  
Mergesort, Quicksort, Binary Search, Binary tree Traversals and related properties, Multiplication of large integers, Stressen’s Matrix Multiplication

**Decrease-and-Conquer**  
Insertion Sort, Depth First and Breadth First Search, Topological sorting, Algorithms for Generating Combinatorial Objects

**Transform-and-Conquer**  
Presorting, Balanced Search Trees, Heaps and Heapsort, Problem Reduction

**Space and Time Tradeoffs**  

**Dynamic Programming**  
Computing a binomial coefficient, Warshall’s and Floyd’s Algorithms, The Knapsack Problem and Memory Functions

**Greedy Technique**  
Prim’s Algorithm, Kruskal’s Algorithm, Dijkstra’s Algorithm, Huffman Trees

**Limitations of Algorithm Power**  
Lower-Bound Arguments, Decision Trees, P, NP and NP-Complete Problems

**Coping with the Limitations of Algorithm Power**  

**Text Books:**
Advanced JAVA Programming

Subject Code: 13MCA42
I.A. Marks: 50
Hours/Week: 4
Exam Hours: 3
Total Hours: 52
Exam Marks: 100

Servlets: 8 Hours

JSP: 8 Hours
Overview of JSP Technology, Need of JSP, Benefits of JSP, Advantages of JSP, Basic syntax, Invoking java code with JSP scripting elements, creating Template Text, Invoking java code from JSP, Limiting java code in jsp, using jsp expressions, comparing servlets and jsp, writing scriptlets, scriptlet example Using Scriptlets to make parts of jsp conditional, using declarations, declaration example. Controlling the Structure of generated servlets: the JSP page directive, import attribute, session attribute, isElgnore attribute, buffer and auto flush attributes, info attribute, errorPage and is errorPage attributes, is Thread safe Attribute, extends attribute, language attribute, Including files and applets in jsp Pages, using java beans components in JSP documents.

Java Beans & Annotations: 6 Hours
Creating Packages, Interfaces, JAR files and Annotations. The core java API package, New java. Lang Sub package, Built-in Annotations. Working with Java Beans. Introspection, Customizers, creating java bean, manifest file, Bean Jar file, new bean, adding controls, Bean properties, Simple properties, Design Pattern events, creating bound properties, Bean Methods, Bean an Icon, Bean info class, Persistence, Java Beans API.

JDBC: 8 Hours
Talking to Database, Immediate Solutions, Essential JDBC program, using prepared Statement Object, Interactive SQL tool. JDBC in Action Result sets, Batch updates, Mapping, Basic JDBC data types, Advanced JDBC data types, immediate solutions.

Swings in Depth: 8 Hours

Introduction to EJB: 5 Hours
Developing your first EJB, preparation, Definitions, naming conventions, convention for the Examples, coding the EJB, the contract, the bean Implementation class, out of Container Testing, Integration Testing.

**Server Side Component Models:** 9 Hours
The Stateless Session Bean, the Stateful Session Bean, the Singleton Session Bean, Message-Driven Beans. EJB and PERSISTENCE. Persistence Entity manager Mapping Persistence objects, Entity Relationships.

**Text Books:**
2. Java 6 Programming Black Book, Dreamtech Press. 2012 (Chapter 17,18,19,20,21,22,27,28,29,30).
3. Andrew LeeRubinger, Bill Burke. Developing Enterprise Java Components. Enterprise JavaBeans 3.1.O’rei!ly. (Chapter 1,2,3,4,5,6,7,8,9,10,11).

**Reference Books:**

---

**Advanced Web Programming**

**Subject Code:** 13MCA43 **I.A. Marks :** 50
**Hours/Week :** 4 **Exam Hours: 3**
**Total Hours :** 52 **Exam Marks: 100**

1. **Programming in Perl** 7 Hours
   Origins and uses of Perl, Scalars and their operations, Assignment statements and simple input and output, Control statements, Fundamentals of arrays, Hashes, References, Functions, Pattern matching, File input and output; Examples.

2. **CGI Scripting** 6 Hours
   What is CGI? Developing CGI Applications, Processing CGI, Introduction to CGI.pm, CGI.pm methods, Creating HTML Pages Dynamically, Using CGI.pm – An Example, Adding Robustness, Carp, Cookies

3. **Building Web Applications with Perl** 5 Hours
   Uploading files, Tracking users with Hidden Data, Using Relational Databases, using libwww,

4. **Introduction to PHP** 8 Hours
   Origins and uses of PHP, Overview of PHP, General syntactic characteristics, Primitives, operations and expressions, Output, Control statements, Arrays, Functions, Pattern matching, Form handling, Files

5. **Building Web applications with PHP** 6 Hours
   Tracking users, cookies, sessions, Using Databases, Handling XML.
6. Introduction to Ruby
Origins and uses of Ruby, Scalar types and their operations, Simple input and output, Control statements, Arrays, Hashes, Methods, Classes, Code blocks and iterators, Pattern matching.

7. Introduction to Rails
Overview of Rails, Document requests, Processing forms, Rails applications with Databases, Layouts.

8. Introduction web 2.0,
What is Web 2.0?, Folksonomies and Web 2.0, Software As a Service (SaaS), Data and Web 2.0, Convergence, Iterative development, Rich User experience, Multiple Delivery Channels, Social Networking.

9. Web Services
Web Services: SOAP, RPC Style SOAP, Document style SOAP, WSDL, REST services, JSON format, What is JSON?, Array literals, Object literals, Mixing literals, JSON Syntax, JSON Encoding and Decoding, JSON versus XML.

Text Books:
3. Francis Shanahan: Mashups, Wiley India 2007(Chapters 1, 6)

Reference Books:

Advanced Computer Networks

Subject Code: 13MCA441 IA Marks : 50
Hours/Week: 4 Exam Hours: 3
Total Hours: 52 Exam Marks: 100

Review of Network Models
Layered tasks; The OSI model and layers in the OSI model; TCP / IP protocol suite; Addressing

SONET / SDH
Architecture; SONET layers; SONET frames; STS multiplexing; SONET networks; Virtual tributaries

Frame Relay and ATM
Frame relay; ATM and ATM LANs

IPv6, Address Mapping and Error Reporting
IPv6: Advantages, Packet format, and Extension headers; Transition from IPv4 to IPv6: Dual stack, Tunneling, and Header translation; Address mapping: ARP, RARP, BOOTP, and DHCP; Error reporting: ICMP.
Multicast Routing Protocols  
Unicast, multicast and broadcast; Applications; Multicasting routing; Routing protocols.

SCTP  
SCTP services; SCTP features; Packet format; An SCTP association; Flow control; Error control; Congestion control.

Congestion Control and Quality of Service  
Data traffic; Congestion and congestion control; Congestion control in TCP, Frame relay; Quality of Service; Techniques to improve QoS; Integrated services; Differentiated services

Multimedia  
Digitizing audio and video; Audio and video compression; Streaming stored audio / video; Streaming live audio / video; Real-time interactive audio / video; RTP; RTCP; VoIP.

Mobile Ad-Hoc Networks, Wireless Sensor Networks  
Overview of wireless ad-hoc networks; Routing in ad-hoc networks; Routing protocols for ad-hoc networks; Security of ad-hoc networks; Sensor networks and protocol structures; Communication energy model; Clustering protocols; Routing protocols; Zigbee technology and IEEE 802.15.4

Text Books:
(Chapters 2, 17, 18, 20.3, 20.4, 21.1, 21.2, 22.4, 23.4, 24, 29) 
(Chapters 19, 20 excluding 20.5)

References:

Data Warehousing and Data Mining

Data Warehousing and OLAP  
Data Warehouse basic concepts, Data Warehouse Modeling, Data Cube and OLAP

Data Mining  
Introduction, What is Data Mining, Motivating Challenges, Data Mining Tasks, Which technologies are used, which kinds of applications are targeted by Data Mining
Data Mining-Which type of data 6 Hours
Types of Data, Data Preprocessing, Measures of Similarity and Dissimilarity, Data Mining Applications

Association Analysis: Basic Concepts and Algorithms 8 Hours
Frequent Itemset Generation, Rule Generation, Compact Representation of Frequent Itemsets, Alternative methods for generating Frequent Itemsets, FP Growth Algorithm, Evaluation of Association Patterns

Classification 12 Hours

Clustering Techniques 8 Hours
Overview, Features of cluster analysis, Types of Data and Computing Distance, Types of Cluster Analysis Methods, Partitional Methods, Hierarchical Methods, Density Based Methods, Quality and Validity of Cluster Analysis

Outlier Analysis 4 Hours
Outlier detection methods, Statistical Approaches, Clustering based applications, Classification based approached

Text Books:

Reference Books:
2. Jiawei Han and Micheline Kamber: Data Mining - Concepts and Techniques, 2nd Edition, Morgan Kaufmann Publisher, 2006.

Mobile Computing and Wireless Communications

Subject Code: 13MCA443
IA Marks: 50
Hours/Week: 04
Exam Hours: 03
Total Hours: 52
Exam Marks: 100

Mobile Computing Architecture: 6 Hours
Types of Networks, Architecture for Mobile Computing, 3-tier Architecture, Design Considerations for Mobile Computing,

Wireless Networks – 1: GSM and SMS 7 Hours
Global Systems for Mobile Communication (GSM and Short Service Messages (SMS): GSM Architecture, Entities, Call routing in GSM, PLMN Interface, GSM Addresses and Identities, Network Aspects in GSM, Mobility Management, GSM Frequency allocation.
Introduction to SMS, SMS Architecture, SM MT, SM MO, SMS as Information bearer, applications

**Wireless Networks – 2: GPRS**  
6 Hours  
GPRS and Packet Data Network, GPRS Network Architecture, GPRS Network Operations, Data Services in GPRS, Applications for GPRS, Billing and Charging in GPRS

**Wireless Networks – 3: CDMA, 3G and WiMAX**  
7 Hours  
Spread Spectrum technology, IS-95, CDMA versus GSM, Wireless Data, Third Generation Networks, Applications on 3G, Introduction to WiMAX.

**Mobile Client**  
6 Hours  
Moving beyond desktop, Mobile handset overview, Mobile phones and their features, PDA, Design Constraints in applications for handheld devices. Mobile IP: Introduction, discovery, Registration, Tunneling, Cellular IP, Mobile IP with IPv6

**Mobile OS and Computing Environment**  
7 Hours  

**Building, Mobile Internet Applications**  
6 Hours  
Thin client: Architecture, the client, Middleware, messaging Servers, Processing a Wireless request, Wireless Applications Protocol (WAP) Overview, Wireless Languages: Markup Languages, HDML, WML, HTML, cHTML, XHTML, VoiceXML.

**J2ME**  
7 Hours  
Introduction, CDC, CLDC, MIDP; Programming for CLDC, MIDlet model, Provisioning, MIDlet life-cycle, Creating new application, MIDlet event handling, GUI in MIDP, Low level GUI Components, Multimedia APIs; Communication in MIDP, Security Considerations in MIDP.

**Text Books:**

**Reference Books:**
Software Testing and Practices

Sub. Code: 13MCA444        IA Marks-50
Hrs/Week: 4                      Exam Hours:03
Total Hours: 52            Exam Marks: 100

Basics of Software Testing 7 hours
Humans, Errors and Testing, Software Quality; Requirements, Behavior and Correctness, Correctness Vs Reliability; Testing and Debugging; Test Metrics; Software and Hardware Testing; Testing and Verification; Defect Management; Execution History; Test Generation Strategies; Static Testing; Test Generation from Predicates.

Basic Principles 4 hours
Sensitivity, Redundancy, Restriction, Partition, Visibility and Feedback

Test Analysis Activities within a Software Process 3 Hours
The Quality Process, Planning and Monitoring Quality goals, Dependability Properties; Analysis; Testing Improving the Process, Organizational Factors

Finite Models, Dependence and Dataflow Models 8 Hours
Overview, Finite abstraction of Behavior; Control Flow Graphs; Finite State Machines, Definition-Use Pairs; Data Flow Analysis; Cluster Analysis; Live and Avail; From Execution to Conservative Flow Analysis; Data flow analysis with Arrays and Pointers; Inter-Procedural Analysis.

Test Case Selection and Adequacy 3 Hours
Test Specification and cases. Adequacy Criteria, Comparing Criteria

Functional Testing, Combinatorial Testing 6 Hours
Random VS Partition Testing, Testing Strategies; a systematic Approach, Choosing a suitable approach, Category-Partition Testing, Pairwise combination testing, Catalog based testing.

Structural Testing, Data flow testing 6 Hours
Statement and Branch testing, Condition testing, Path testing, Procedure call testing, Comparing Structural testing Criteria; The infeasibility problem, Definition-Use Association; Data flow testing criteria, Data flow coverage with Complex Structures, The infeasibility problem.

Model Based Testing 3 Hours
Deriving test cases from finite State Machines; Testing decision structure; Deriving Test cases from Control, Data Flow Graphs and Grammars.

Fault Based Testing 6 Hours
Assumptions in fault-based testing, Mutation Analysis, Fault-based Adequacy Criteria; Variations on mutation Analysis; From Test case specification to Test Cases, Scaffolding, Generic vs specific Scaffolding, Test Oracles, Self checks as oracles, Capture and Replay.

Planning and Monitoring the Process, Documenting Analysis and Test 6 Hours
Quality and Process, Test and Analysis strategies and plans, Risk Planning, Monitoring the Process, Improving the process, The quality team, Organizing documents, Test strategy document, Analysis and test plan, Test design specifications documents, Test and analysis reports.
Text Books
1. Adithya P. Mathur “Foundations of Software Testing – Fundamental Algorithms and Techniques”, Pearson Education India, 2011 (Listed topics only from Chapter 1.1 to 1.12 and 2.7)
2. Mauro Pezze, Michael Young, Software testing and Analysis- Process, Principles and Techniques, Wiley India, 2012 (Chapters 3,4,5,6,9,10,11,12,13,14,16,17,20,24)

Reference Books

Theory of Computation (Finite Automata and Formal Languages)

Subject Code: 13MCA445
IA Marks: 50
Hours/Week: 04
Exam Marks: 100
Total Hours: 52
Exam Hours: 03

Introduction and Finite Automata: 10 Hours
What is (not) a computer, The idea of computing, Computing Machines and Languages, What is the Science of Computing, Programming, Data Structures, Algorithms and Science, Birth of Science computing, Computability, Undecidability, Intractability and Intelligence, Why Study Science computing and Key Ideas, Automata- The idea of computing Machine, Automata Definition, Constructing Simple Automata, Handling End Condition, Handling Reject States, A Step-by-Step model for constructing Automata, States as Memory, Why Finite number of states, Constructing more complex Automata, Mantras for constructing Automata, Limitations of Finite Automata, Automata with Combinatorial States

NFA and Regular Expression 7 Hours
The idea of Non-Determinism, Constructing Non-Deterministic Automata, Eliminating Non-Deterministic: converting NFA to DFA, Jumping States without Input, A method for minimizing Automata, Finite State Transducers, The idea of formal languages, Languages of Automata, Regular Expression, Constructing Regular Expressions, Converting Regular Expressions to Automata, Equivalence of Regular Expressions, Method for Constructing Regular Expressions, Regular Expressions in Practice

Regular Grammars and Languages 7 Hours
The idea of Grammar, The ideas of parsing and Derivation, Grammars for Regular Languages, Constructing Regular Grammars, converting automata to regular grammars, converting regular grammars to automata, constructing regular grammars: mantras, Closure properties, Answering questions about regular languages, Why are some languages not regular, The Pigeonhole Principle and Pumping Lemma, Using Pumping Lemma an Adversarial Game.

Context Free Grammars 7 Hours
The idea and nature of context free grammar, Constructing Context free grammars (LGs and Non LGs), Introduction to Parsing, Ambiguity and Eliminating ambiguity, The idea of Chomsky normal form, Converting to Chomsky normal form, The ideas of Greibach Normal form, Simple Linear and other grammars.
Pushdown Automata and Nature of Context Free Languages  
7 Hours
Machines for Context Free Languages, Adding Memory: Why Stack Behavior, Constructing PDAs, Constructing CFGs to PDAs, Converting PDAs to CFGs, Non-determinism in PDAs, The CFL-CFG-PDA Triad, Closure Properties, Union of CFLs, Answering Questions about CFLs, Why are some languages not context-free, The pumping lemma for context free languages.

Turing Machines  
8 Hours

The Chomsky Hierarchy  
6 Hours
Languages, Grammars and Machines, Recursively Enumerable Languages, Counting Alphabets, Languages and Computing Machines, The idea of Enumeration, The idea of Diagonalization, The ideas of Acceptance and Membership, Recursive Languages, Context Sensitive Languages and Grammars, The ideas of context, Other Grammars and Automata, Linear and Deterministic Context-Free Languages.

Text Books:

Reference Books:
2. Introduction to Automata Theory, Languages, and Computation, Addison Weisly Publishing company, 2010

CRYPTOGRAPHY AND NETWORK SECURITY

SubCode: I3MCA451  
IAMarks: 50
Hrs/Week: 4  
Exam Hours: 3
Total Hours:52  
Exam Marks:100

Planning for Security:
6 Hours
Introduction, Information Security Policy, Standards and Practices; The Information Security Blue Print; Contingency plans and a model for contingency plan.

Security Technology:
10 Hours
Introduction; Physical designs; Firewalls; Protecting Remote Connections Introduction; Intrusion Detection (IDS); Honey Pots, Honey Nets and Padded cell systems; Scanning and Analysis Tools.

Introduction to Cryptography:
Traditional Symmetric – Key Cipher and Asymmetric – Key Cryptography  
10 Hours
Introduction to Asymmetric – Key Cryptography, RSA Cryptosystems, Rabin ElGamal, Elliptic Curve Cryptoystems.

**Data Encryption Standard (DES) and Advanced Encryption Standard (AES)** 8 Hours
Introduction, DES Structure and Analysis, Security of DES, Multiple DES; Introduction to AES, Transformation, Key Expressions, AES Ciphers

**Message Integrity, Message Authentication and Key Management** 6 Hours
Message Integrity, Random Oracle Model, Message Authentication, Symmetric-Key Distribution, Kerberos, Symmetric-Key Agreement, Public-Key Distribution, Hijacking

**Security at the Application Layer : PGP and S/MIME** 4 Hours
E-mail, PGP, S/MIME

**Security at the Transport Layer and Security at the Network Layer IPSec** 8 Hours

Text Books:

Network Management

**Network Management**

**Requirements for the Management of Networked Systems** 05 Hours
Management Scenarios, Management functions, Organizational aspects of Management, Time aspects of Management

**IP Network Management** 07 Hours
IP-Based Service Implementation and Network Management 08 Hours

Network Management Architecture 06 Hours

SLA and Network Monitoring 05 Hours
Passive and Active Network Monitoring, Passive Network Monitoring, Active Network Monitoring.

MPLS Network Management: AN Introduction 05 Hours
A brief Introduction to MPLS, MPLS Applications, Key Aspects of MPLS Network Management, Management Information Base Modules for MPLS.

MPLS Management Interfaces 05 Hours
The basics of Management Interfaces, Command line interface, CORBA, XML, Bulk File Transfer, Simple Network Management Protocol

Optical Networks: Control and Management 05 Hours
Network Management functions, Optical Layer Services and Interfacing, Layers within the Optical Layer, Multivendor Interoperability, Performance and Fault Management, Configuration

Web-Based Management 06 Hours

Text Books:
1. Network Management- Know it all by Adrian Farrel, Elsevier publications. Chapter 1-8
2. Network Management- Principles and Practice, Mani Subramaniam, Pearson Education. Chapter 14

Reference Books:
1. Network Management, Morris, Pearson Education
2. Practical Guide to SNMPv3 and Network Management, David Zeltserman, PHI.
NOSQL

Subject Code: 13MCA453  I.A. Marks : 50
Hours/Week : 4       Exam Hours: 3
Total Hours : 52     Exam Marks: 100

Introduction to NoSQL   6 Hours
Definition of NoSQL, History of NoSQL and Different NoSQL products, Exploring MondoDB Java/Ruby/Python, Interfacing and Interacting with NoSQL.

NoSQL Basics   12 Hours
NoSQL Storage Architecture, CRUD operations with MongoDB, Querying, Modifying and Managing NoSQL Datastores, Indexing and ordering datasets (MongoDB/CouchDB/Cassandra)

Advanced NoSQL   8 Hours
NoSQL in CLOUD, Parallel Processing with MapReduce, BigData with Hive

Working with NoSQL   10 Hours
Surveying Database Internals, Migrating from RDBMS to NoSQL, WebFrameworks and NoSQL, using MySQL as a NoSQL

Developing Web Application with NoSQL and NoSQL Administration   16 Hours
Php and MongoDB, Python and MongoDB, Creating Blog Application with PHP, NoSQL Database Administration.

Text Books
   (Chapter 1,2,3,4,5,6,7,8,9,10,11,12,13.15)
   (Chapter 6,7,8,9)
# Software Architectures

**Subject Code:** 13MCA454  
**I. A. Marks:** 50  
**Hours/Week:** 4  
**Exam Hours:** 3  
**Total Hours:** 52  
**Exam Marks:** 100

## Introduction  
06 Hours  
What software architecture is and what it is not; Architectural Structures and views; Architectural patterns; What makes a “good” architecture? Why is software important?

## Context of Software Architecture  
04 Hours  
Technical Context; Project life-cycle context; Business context; Professional context; Stake holders; How is Architecture influenced? What Do Architecture influence?

## Understanding Quality Attributes  
12 Hours  
Architecture & Requirements; Functionality; quality attribute considerations; Specifying and achieving Quality attribute requirements; Guiding quality design decisions; Availability; Interoperability; Modifiability; Performance; Security; Testability; Usability

## Quality Attribute modeling and Analysis  
06 Hours  
Modeling Architecture to enable quality attribute analysis; Quality attribute check lists; Through experiments and Back-of-the envelope analysis; Experiments; Simulations and prototypes; Analysis at different stages of the life cycle

## Architecture and requirements  
06 Hours  
Gathering ASRs from requirements documents; ASRs by interviewing stake holders; ASRs by understanding the business; capturing ASRs in a utility tree; Typing the methods together

## Designing an Architecture  
03 Hours  
Design strategy; the attribute driven design methods; the steps of ADD

## Documenting Software Architecture  
06 Hours  
Uses and Audiences for architecture documentation; Notations, View and Behavior; Documentation and quality attributes

## Architecture, Implementation & Testing  
03 Hours  
Architecture and implementation; Architecture and testing

## Architectural Patterns  
06 Hours  
Introduction to patterns; From Mud to structure; Layers; Pipes and filters; Blackboard; Distributed systems; Broker; Interactive systems; Model-view-control; Presentation-abstraction-control; Adaptable systems; Microkernel
Enterprise Resource Planning

Subject Code: 13MCA455

I.A. Marks: 50

Hours/Week: 4

Exam Hours: 3

Total Hours: 52

Exam Marks: 100

UNIT I INTRODUCTION TO ERP

Overview, Benefits of ERP, ERP and Related Technologies, Business Process Reengineering, Data Warehousing, Data Mining, On–line Analytical Processing, Supply Chain Management.

UNIT II ERP IMPLEMENTATION


UNIT III BUSINESS MODULES


UNIT IV ERP MARKET


UNIT V ERP – PRESENT AND FUTURE

Turbo Charge the ERP System, EIA, ERP and E–Commerce, ERP and Internet, Future Directions in ERP.

Text Books


Reference Books

Implement the following using C/C++ Language.

1. Implement Recursive Binary search and Linear search and determine the time required to search an element. Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n.
2. Sort a given set of elements using the Heapsort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
3. Sort a given set of elements using Merge sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
4. Obtain the Topological ordering of vertices in a given digraph.
5. Implement 0/1 Knapsack problem using dynamic programming.
6. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
7. Sort a given set of elements using Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
8. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
9. Print all the nodes reachable from a given starting node in a digraph using BFS method.
10. Check whether a given graph is connected or not using DFS method.
11. Find a subset of a given set S = \{s_1,s_2,\ldots,s_n\} of n positive integers whose sum is equal to a given positive integer d. For example, if S = \{1, 2, 5, 6, 8\} and d = 9 there are two solutions \{1,2,6\} and \{1,8\}. A suitable message is to be displayed if the given problem instance doesn't have a solution.
    b. Find the Binomial Co-efficient using Dynamic Programming.
13. Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.
14. a. Implement Floyd’s algorithm for the All-Pairs-Shortest-Paths problem.
    b. Compute the transitive closure of a given directed graph using Warshall's algorithm.
15. Implement N Queen's problem using Back Tracking.

Note: In the examination questions must be given based on lots.

Advanced Java Programming Laboratory

Subject Code: 13MCA47
Hours/Week: 3
Total Hours: 42

1. Write a JAVA Servlet Program to implement a dynamic HTML using Servlet (user name and Password should be accepted using HTML and displayed using a Servlet).
2. Write a JAVA Servlet Program to Auto Web Page Refresh (Consider a webpage which is displaying Date and time or stock market status. For all such type of pages, you would need to
refresh your web page regularly; Java Servlet makes this job easy by providing refresh automatically after a given interval).
3. Write a JAVA Servlet Program to implement and demonstrate get() and Post methods(Using HTTP Servlet Class).
4. Write a JAVA Servlet Program using cookies to remember user preferences.
5. a. Write a JAVA JSP Program to implement verification of a particular user login and display a Welcome page.
   b. Write a JSP program to demonstrate the import attribute.
6. a. Write a JAVA JSP Program which uses jsp: include action to display a Webpage.
   b. Write a JAVA JSP Program which uses <jsp:plugin> tag to run a applet.
7. Write a JAVA JSP Program to get student information through a HTML and create a JAVA Bean.
8. Class, populate Bean and display the same information through another JSP.
9. Write a JAVA Program to insert data into Student DATA BASE and retrieve info based on particular queries(For example update, delete, search etc…).
10. Write a java program using swings to design menu bar and menu items.
11. An EJB application that demonstrates Session Bean (with appropriate business logic).
12. An EJB application that demonstrates MDB (with appropriate business logic).
13. An EJB application that demonstrates persistence (with appropriate business logic).

Object-Oriented Modeling and Design Patterns

Subject Code: 13MCA51     I.A. Marks : 50
Hours/Week : 4             Exam Hours: 3
Total Hours : 52            Exam Marks: 100

PART – A

1. Introduction, Modeling Concepts   3 Hrs
What is Object Orientation? What is OO development? OO themes; Evidence for usefulness of OO development; OO modeling history. Modeling as Design Technique: Modeling; abstraction; The three models.

2. Class Modeling and Advanced Class Modeling:   7 Hrs
Object and class concepts; Link and associations concepts; Generalization and inheritance; A sample class model; Navigation of class models; Practical tips. Advanced object and class concepts; Association ends; N-array associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived data; Packages; Practical tips

3. State Modeling and Advanced State Modeling   6 Hrs
State Modeling: Events, States, Transitions and Conditions; State diagrams; State diagram behavior; Practical tips. Advanced State Modeling: Nested state diagrams; Nested states; Signal generalization; Concurrency; A sample state model; Relation of class and state models; Practical tips.

4. Interaction Modeling and Advanced Interaction Modeling   4 Hrs
Interaction Modeling: Use case models; Sequence models; Activity models. Use case relationships; Procedural sequence models; Special constructs for activity models.
5. Process Overview, System Conception 3 Hrs
Process Overview: Development stages; Development life cycle. System Conception: Devising a system concept; Elaborating a concept; Preparing a problem statement.

6. Domain Analysis and Application Analysis 8 Hrs
Overview of analysis; Domain class model; Domain state model; Domain interaction model; Iterating the analysis. Application Analysis: Application interaction model; Application class model; Application state model; Adding operations.

7. System Design and Class Design 7 Hrs
Overview of system design; Estimating performance; Making a reuse plan; Breaking a system into sub-systems; Identifying concurrency; Allocation of sub-systems; Management of data storage; Handling global resources; Choosing a software control strategy; Handling boundary conditions; Setting the trade-off priorities; Common architectural styles; Architecture of the ATM system as the example. Class Design: Overview of class design; Bridging the gap; Realizing use cases; Designing algorithms; Recursing downwards, Refactoring; Design optimization; Reification of behavior; Adjustment of inheritance; Organizing a class design; ATM example.

8. Patterns 4 Hrs
What is a pattern and what makes a pattern? Pattern categories; Relationships between patterns; Pattern description.

9. Design Patterns 10 Hrs
Introduction, structural decomposition, Organization of work, Access control; Communication Patterns: Forwarder-Receiver; Client-Dispatcher-Server; Publisher-Subscriber; Management Patterns: Command processor; View Handler;

Text Books:
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2006. (Chapters 1, 3)

Reference Books:
5. E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns-Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.
Introduction 8 Hours
When simulation is the appropriate tool and when it is not appropriate; Advantages and disadvantages of Simulation; Areas of application; Systems and system environment; Components of a system; Discrete and continuous systems; Model of a system; Types of Models; Discrete-Event System Simulation; Steps in a Simulation Study. Simulation example: Simulation of queuing systems in a spreadsheet.

General Principles, Simulation Software 6 Hours

Statistical Models in Simulation 6 Hours
Review of terminology and concepts; Random Variables, Probability Distribution, Probability distribution function, Useful statistical models; Discrete distributions; Continuous distributions; Poisson process; Empirical distributions.

Queuing Models 6 Hours
Characteristics of queuing systems; Queuing notation; Long-run measures of performance of queuing systems; Steady-state behavior of M/G/1 queue; Networks of queues;

Random-Number Generation, Random-Variate Generation 8 Hours
Properties of random numbers; Generation of pseudo-random numbers; Techniques for generating random numbers; Tests for Random Numbers Random-Variate Generation: Inverse transform technique; Acceptance-Rejection technique; Special properties.

Input Modeling 6 Hours
Data Collection; Identifying the distribution with data; Parameter estimation; Goodness of Fit Tests; Fitting a non-stationary Poisson process; Selecting input models without data; Multivariate and Time-Series input models, uniformity and independence, Chi-Square test, K-S Test

Estimation of Absolute Performance & Computer System Simulation 12 Hours
Types of simulations with respect to output analysis; Stochastic nature of output data; Absolute measures of performance and their estimation; Output analysis for terminating simulations; Output analysis for steady-state simulations. Verification, Calibration, and Validation; Optimization: Model building, verification and validation; Verification of simulation models; Calibration and validation of models.

Text Books:
   (Listed topics only from Chapters1 to 12)
Reference Books:

Programming Using C#.NET

Subject: 13MCA53
IA Marks: 50
Hours/Week: 04
Exam Hours: 03
Total Hours: 52
Exam Marks: 100

Getting started with .NET Framework 4.0
04 Hours

Introducing C#
06 Hours
Need of C#, C# Pre-processor Directives, Creating a Simple C# Console Application, Identifiers and Keywords. Data Types, Variables and Constants: Value Types, Reference Types, Type Conversions, Boxing and Unboxing, Variables and Constants. Expression and Operators: Operator Precedence, Using the ?? (Null Coalescing) Operator, Using the :: (Scope Resolution) Operator and Using the is and as Operators. Control Flow statements: Selection Statements, Iteration Statements and Jump Statements.

Namespaces, Classes, Objects and Structures
06 Hours

Object-Oriented Programming
05 Hours

Delegates and Events and Exception Handling
05 Hours
Graphical User Interface with Windows Forms  
Introduction, Windows Forms, Event Handling: A Simple Event-Driven GUI, Visual Studio Generated GUI Code, Delegates and Event-Handling Mechanism, Another Way to Create Event Handlers, Locating Event Information. Control Properties and Layout, Labels, TextBoxes and Buttons, GroupBoxes and Panels, CheckBoxes and RadioButtons, ToolTips, Mouse-Event Handling, Keyboard-Event Handling. Menus, MonthCalendar Control, DatePicker Control, LinkLabel Control, ListBox Control, CheckedListBox Control, ComboBox Control, TreeView Control, ListView Control, TabControl Control and Multiple Document Interface (MDI) Windows.

Data Access with ADO.NET  

Web App Development with ASP.NET  

Text Books:  
1. .NET 4.0 Programming (6-in-1), Black Book, Kogent Learning Solutions Inc., Wiely-Dream Tech Press. (Chapters: 1,10,11,12,13,14 and 19).  

References Books:  
2. Bart De Smet: C# 4.0 Unleashed, Pearson Education- SAMS Series.  
### Mobile and Adhoc Sensor Networks

**Subject Code:** 13MCA541  \hspace{1cm} **I.A. Marks:** 50  
**Hours/Week:** 4  \hspace{1cm} **Exam Hours:** 3  
**Total Hours:** 52  \hspace{1cm} **Exam Marks:** 100

**Unit I**

**Unit II**
Topology Control in Wireless Ad Hoc Networks, Broadcasting and Activity Scheduling in Ad Hoc Networks, Location Discovery, Mobile Ad Hoc Networks (MANETs): Routing Technology for Dynamic, Wireless Networking, Routing Approaches in Mobile Ad Hoc Networks.  \hspace{1cm} **10 Hours**

**Unit III**

**Unit IV**

**Text Books and References**

### Parallel Computing

**Subject Code:** 13MCA542  \hspace{1cm} **I.A. Marks:** 50  
**Hours/Week:** 4  \hspace{1cm} **Exam Hours:** 3  
**Total Hours:** 52  \hspace{1cm} **Exam Marks:** 100

**SCALABILITY AND CLUSTERING**  \hspace{1cm} **10 Hours**

**ENABLING TECHNOLOGIES**  \hspace{1cm} **12 Hours**
SYSTEM INTERCONNECTS 10 Hours
Basics of Interconnection Networks, Network Topologies and Properties, Buses, Crossbar and Multistage Switches, Software Multithreading, Synchronization Mechanisms.

PARALLEL PROGRAMMING 10 Hours
Paradigms And Programmability, Parallel Programming Models, Shared Memory Programming.

MESSAGE PASSING PROGRAMMING 10 Hours

TEXT BOOK

REFERENCES

Multimedia Systems

Subject Code: 13MCA543  I.A. Marks : 50
Hours/Week : 4           Exam Hours: 3
Total Hours : 52         Exam Marks: 100

Introduction, Media and Data Streams, Audio Technology 7 Hours
Multimedia Elements; Multimedia Applications; Multimedia Systems Architecture; Evolving Technologies for Multimedia Systems; Defining Objects for Multimedia Systems; Multimedia Data Interface Standards; The need for Data Compression; Multimedia Databases.
Sound: Frequency, Amplitude, Sound Perception and Psychoacoustics; Audio Representation on Computers; Three Dimensional Sound Projection; Music and MIDI Standards; Speech Signals; Speech Output; Speech Input; Speech Transmission.

Graphics and Images, Video Technology, Computer-Based Animation 7 Hours
Capturing Graphics and Images Computer Assisted Graphics and Image Processing; Reconstructing Images; Graphics and Image Output Options.
Basics; Television Systems; Digitalization of Video Signals; Digital Television; Basic Concepts; Specification of Animations; Methods of Controlling Animation; Display of Animation; Transmission of Animation; Virtual Reality Modeling Language.

Data Compression 12 Hours
Storage Space; Coding Requirements; Source, Entropy, and Hybrid Coding; Basic Compression Techniques; JPEG: Image Preparation, Lossy Sequential DCT-based Mode, Expanded Lossy DCT-based Mode, Lossless Mode, Hierarchical Mode.
H.261 (Px64) and H.263: Image Preparation, Coding Algorithms, Data Stream, H.263+ and H.263L; MPEG: Video Encoding, Audio Coding, Data Stream, MPEG-2, MPEG-4, MPEG-7; Fractal Compression.

**Optical Storage Media**  
6 Hours  
History of Optical Storage; Basic Technology; Video Discs and Other WORMs; Compact Disc Digital Audio; Compact Disc Read Only Memory; CD-ROM Extended Architecture; Further CD-ROM-Based Developments; Compact Disc Recordable; Compact Disc Magneto-Optical; Compact Disc Read/Write; Digital Versatile Disc.

**Content Analysis**  
6 Hours  
Simple Vs. Complex Features; Analysis of Individual Images; Analysis of Image Sequences; Audio Analysis; Applications.

**Data and File Format Standards**  
7 Hours  
Rich-Text Format; TIFF File Format; Resource Interchange File Format (RIFF); MIDI File Format; JPEG DIB File Format for Still and Motion Images; AVI Indeo File Format; MPEG Standards; TWAIN

**Multimedia Application Design**  
7 Hours  
Multimedia Application Classes; Types of Multimedia Systems; Virtual Reality Design; Components of Multimedia Systems; Organizing Multimedia Databases; Application Workflow Design Issues; Distributed Application Design Issues.

**Text Books:**  
   (Chapters 2, 3, 4, 5, 6, 7, 8, 9)  
   (Chapters 1, 3, 7)

**Reference Books:**  

**PATTERN RECOGNITION**

Subject Code : 13MCA544  
IA Marks : 50
No. of Lecture Hrs/Week : 04  
Exam Hours : 03
Total no. of Lecture Hrs. : 52  
Exam Marks : 100

**UNIT - 1**  
INTRODUCTION: Applications of pattern recognition, statistical decision theory, image processing and analysis.  
6 Hours

**UNIT - 2**  
PROBABILITY: Introduction, probability of events, random variables, Joint distributions and densities, moments of random variables, estimation of parameters from samples, minimum risk estimators.  
6 Hours
UNIT - 3
STATISTICAL DECISION MAKING: Introduction, Baye’s Theorem, multiple features, conditionally independent features, decision boundaries, unequal costs of error, estimation of error rates, the leavingone-out technique. Characteristic curves, estimating the composition of populations.  
7 Hours

UNIT - 4
NONPARAMETRIC DECISION MAKING: Introduction, histograms, Kernel and window estimators, nearest neighbor classification techniques, adaptive decision boundaries, adaptive discriminate Functions, minimum squared error discriminate functions, choosing a decision making technique.  
6 Hours

UNIT - 5
CLUSTERING: Introduction, hierarchical clustering, partitional clustering.  
6 Hours

UNIT - 6
ARTIFICIAL NEURAL NETWORKS: Introduction, nets without hidden layers. nets with hidden layers, the back Propagation algorithms, Hopfield nets, an application.  
7 Hours

UNIT - 7
PROCESSING OF WAVEFORMS AND IMAGES: Introduction, gray level sealing transfoniiations, equalization, geometric image and interpolation, Smoothing, transformations, edge detection, Laplacian and sharpening operators, line detection and template matching.  
7 Hours

UNIT-8
IMAGE ANALYSIS: Introduction, Scene segmentation and labeling, counting objects, perimeter measurement, Hough Tranforms, Morphological Operations, texture, Fourier transforms, The classification of White Blood Cells  
7 Hours

TEXT BOOKS:

REFERENCE BOOKS
1. “Pattern recognition (Pattern recognition a scene analysis)” Duda and Hart.
Service Oriented Architectures (SOA)

Sub. Code: 13MCA545        IA Marks-50
Hrs/Week: 4                      Exam Hours:03
Total Hours: 52            Exam Marks: 100

Introduction to SOA, Evolution of SOA 6 hours
Fundamentals of SOA, Common characteristics of contemporary SOA, Common tangible benefits of SOA, A SOA timeline (from XML to Web Services to SOA), The continuing evolution of SOA (standards organizations and Contributing vendors), The roots of SOA (comparing SOA to Past Architectures)

Web Services and Primitives of SOA 6 hours
The Web Services framework, Services (as Web Services), Service Description (with WSDL), Messaging (with SOAP)

Web Services and Contemporary SOA 12 Hours
Message Exchange patterns, Service Activity; Coordination, Atomic Transactions, Business Activities, Orchestration, Choreography, Addressing, Reliable Messaging, Correlation, Policies, Meta data Exchange, Security, Notification and eventing.

Principles of Service – Orientation 7 Hours
Services- Orientation and the enterprise, Anatomy of service-oriented Architecture, Common Principles of Service Orientation; How Service Orientation principles inter relate, Service Orientation and object orientation, Native Web Service support for service orientation principles.

Service Layers 6 Hours
Service Orientation and contemporary SOA, Service Layer Abstraction, Application service layer, Business Service Layer, Orchestration Service Layer, Agnostic Services, Service Layer Configuration scenarios.

Business Process Design 7 Hours
WS-BPEL Language basics, WS-Coordination overview, Service oriented business process redesign, WS-Addressing language basics, Ws-Reliable messaging language basics.

Enterprise Applications 8 Hours

Text Books
1. Thomas Erl: Service Oriented Architecture- Concepts, Technology and Design, Pearson Education, 2013 (listed topics only from Chapters 3,4,5,6,7,8,9,16,17)
2. Shankar Khambhaty, Service Oriented Architecture for Enterprise and Cloud Applications, 2nd Edition, Wiley-India, 2012 (listed topics only from Chapter 5,6)

Reference Books
1. Frank cohen: FastSOA, Elsevier, 2010
Compiler Design

Subject Code: 13MCA546    IA Marks: 50
Hours/ Week: 4     Exam Hours: 3
Total Hours: 52                Exam Marks: 100

Introduction, Lexical analysis  8 Hours
Language processors; The structure of a Compilers; The evolution of programming languages; The science of building a compiler; Applications of Compiler technology; Programming language basics; Lexical analysis: The Role of Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens.

Syntax Analysis - 1  6 Hours
Introduction; Context-free Grammars; Writing a Grammar; Top-down Parsing

Syntax Analysis – 2  6 Hours
Bottom-up Parsing; Introduction to LR Parsing: Simple LR.

Syntax Analysis – 3  6 Hours
More powerful LR parsers; Using ambiguous grammars; Parser Generators.

Syntax-Directed Translation  6 Hours
Syntax-Directed definitions; Evaluation order for SDDs; Applications of Syntax-directed translation; Syntax-directed translation schemes

Intermediate Code Generation  8 Hours
Variants of syntax trees; Three-address code; Types and declarations; Translation of expressions; Type checking; Control flow; Back patching; Switch statements; Intermediate code for procedures.

Run-Time Environments  6 Hours
Storage Organization; Stack allocation of space; Access to non-local data on the stack; Heap management; Introduction to garbage collection

Code Generation  6 Hours
Issues in the design of Code Generator; The Target language; Addresses in the target code; Basic blocks and Flow graphs; Optimization of basic blocks; A Simple Code Generator.

Text Books:
1. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman: Compilers- Principles, Techniques and Tools, 2nd Edition, Addison-Wesley, 2007. (Chapters 1, 3.1 to 3.4, 4, 5.1 to 5.4, 6, 7.1 to 7.5, 8.1 to 8.6)

Reference Books:
CLOUD COMPUTING

Distributed System Models and Enabling Technologies

Computer Clusters for scalable parallel computing
Clustering for massive parallelism: Cluster Development Trends, Design Objective of Computer Clusters, Fundamental Cluster Design issues. Virtual machines and Virtualization of clusters and Data centers: Implementation levels of virtualization: levels of virtualization Implementation, VMM Design requirements and providers, Virtualization support at the OS level, Middleware Support for Virtualization.

Cloud Platform Architecture over Virtualized Data Centers

Public Cloud Platforms

Cloud Programming and Software Environments
Features of Cloud and Grid Platforms: Cloud Capabilities and Platform Features, Traditional Features Common to Grids and Clouds, Data Features and Databases, Programming and Runtime Support. Parallel and Distributed Programming Paradigms: Parallel Computing and Programming Paradigms, MapReduce, Twister and Iterative MapReduce, Hadoop Library from Apache.

Programming Support of Google App Engine
Programming the Google App Engine, Google File System (GFS), Bigtable, Google’s NOSQL system, Chubby, Google’s Distributed Lock service. Programming on Amazon AWS and Microsoft Azure: Programming on Amazon EC2, Amazon Simple Storage Service S3, Amazon Elastic Block Store EBS and SimpleDB, Microsoft Azure programming support. Emerging
Cloud Software Environments: Open Source Eucalyptus and Nimbus, OpenNebula, Sector/Sphere, and OpenStack, Manjrasoft Aneka Cloud and Appliances.

**Ubiquitous Clouds and the Internet of Things** 8 Hours


**Text Book:**

**Reference Books:**

**WEB 2.0 AND RICH INTERNET APPLICATIONS**

SubCode: 13MCA552
Hrs/Week: 4
Total Hours: 52

**Exam Marks:** 100

Building Rich Internet Applications with AJAX 6 Hours

Building Rich Internet Applications with AJAX: Limitations of Classic Web application model, AJAX principles, Technologies behind AJAX, Examples of usage of AJAX, Dynamic web applications through Hidden frames for both GET and POST methods. IFrames, Asynchronous communication and AJAX application model.

**Ajax with XMLHTTP object** 6 Hours

Creating Ajax Applications: An example, Analysis of example ajax.html, Creating the JavaScript, Creating and opening the XMLHttpRequest object, Data download, Displaying the fetched data, Connecting to the server, Adding Server-side programming, Sending data to the server using GET and POST.
Handling multiple XMLHttpRequest objects in the same page, Using two XMLHttpRequest objects, Using an array of XMLHttpRequest objects, Using inner functions, Downloading JavaScript, connecting to Google Suggest, Creating google.php, Downloading from other domains with Ajax, HTML header request and Ajax, Defeating caching, Examples.
Building XML and working with XML in JavaScript, Getting the document element, Accessing any XML element, Handling whitespace in Firefox, Handling cross-browser whitespace,
Accessing XML data directly, Validating XML, Further examples of Rich Internet Applications with Ajax

**Ajax Patterns**

Predictive fetch pattern, Submission throttling pattern, Periodic refresh, Multi stage download, Fall back patterns

**Working with PHP and DOM in Ajax**

Working with PHP server variables, Getting the data in to array format, Wrapping applications in to a single PHP page, Validating input from the user, Validating integers and text, DOM, Appending new elements to a web page using the DOM and Ajax, Replacing elements using the DOM, Handling timeouts in Ajax, Downloading images with Ajax, Example programs.

**Flex – 1: Understanding Flex Environment and Layouts**

Introduction: Understanding Flex Application Technologies, Using Flex Elements, Working with Data Services (Loading Data at Runtime), The Differences between Traditional and Flex Web Applications, Understanding How Flex Applications Work, Understanding Flex and Flash Authoring.

Building Applications with the Flex Framework: Using Flex Tool Sets, Creating Projects, Building Applications, Deploying Applications


**Flex – 2: Working with MXML and ActionScript**

MXML: Understanding MXML Syntax and Structure, Making MXML Interactive
Working with UI Components: Understanding UI Components, Buttons, Value Selectors, Text Components, List-Based Controls, Pop-Up Controls, Navigators, Control Bars
Customizing Application Appearance: Using Styles, Skinning components, Customizing the preloader, Themes, Runtime CSS

**Flex – 3: Working with States**

Flex – 4: Working with Data Models and Data Binding 4 Hours

Working with Data: Using Data Models, Data Binding, Enabling Data Binding for Custom Classes, Data Binding Examples, Building data binding proxies. Validating and Formatting Data: Validating user input, Formatting Data.

Impacts of the Next Generation of the web 3 Hours

Business models for Internet and web, Data Ownership, SAAS, Socialization and cocreation of content.

The Semantic web and Web 2.0 3 Hours

Overview of semantic web, Languages of the Semantic Web, Ontologies, Micro-formats, collaborative tagging and folksonomies.

Text Books:

1. Nicholas C Zakas et al: Professional AJAX, Wiley India, publications, (Chapters 1 to 3)

Reference Books:

1. Chafic Kazon and Joey Lott: Programming Flex 3, O’Reilly, 2011. (Listed topics from Chapters 1 to 8, 12 to 15)
2. Gottfried Vossen and Stephan Hagemann: Unleashing Web 2.0 Elsevier, Inc 2011 (Listed topics from Chapters 5 and 6)
Information Retrieval and Search Engines

Subject Code: 13MCA553
Hours/Week: 4
Total Hours: 52
I.A. Marks: 50
Exam Hours: 3
Exam Marks: 100

UNIT 1  INTRODUCTION
4 Hours
Information Retrieval, Search Engines, Search Engineers.

UNIT 2  ARCHITECTURE OF A SEARCH ENGINE
5 Hours
Architecture, Basic Building Blocks, Text Acquisition, Text Transformation Index Creation, User Interaction, Ranking and Evaluation

UNIT 3  CRAWLS AND FEEDS
6 Hours
Deciding what to search, Crawling the Web, Directory Crawling, Document Feeds, Conversion Problem, Storing the Documents, Detecting Duplicates, removes noise.

UNIT 4  PROCESSING TEXT
8 Hours

UNIT 5  RANKING WITH INDEXES
6 Hours
Abstract Model of Ranking, Inverted indexes, Compression, Entropy and Ambiguity, Delta Encoding, Bit-aligned codes, Auxiliary Structures, Index Construction, Query Processing.

UNIT 6  QUERIES AND INTERFACES
5 Hours
Information Needs and Queries, Query Transformation and Refinement, Showing the Results Cross-Language Search.

UNIT 7  RETRIEVAL MODELS
12 Hours
Overview of Retrieval Models, Boolean Retrieval, The Vector Space Model, Probabilistic Models, Information Retrieval as Classification, BM25 Ranking Algorithm, Complex Queries and Combining Evidence, Web Search, Machine Learning and Information Retrieval.

UNIT 8  EVALUATING SEARCH ENGINES
6 Hours
The Evaluation Corpus, Logging, Effectiveness Metrics, Recall and Precision, Averaging and Interpolation, Efficiency Metrics, Training, Testing, and Statistics

Text Books and References

Introduction to fuzzy set theory 8 hours
Probabilistic reasoning, Fuzzy sets, mathematics of fuzzy set theory, operations on fuzzy sets, comparison of fuzzy and crisp set theory.

Fuzzy mapping 6 hours
One to one mapping, max-min principle, extension principle, implication rules – mamdani implications.

Membership functions 8 hours
Universe of discourse, mapping inside fuzzy domain, fuzzy membership mapping methods, application to real world problems.

Fuzzy knowledge based systems: 8 hours
Fuzzification, Fuzzy knowledge base, rule base, Data base for fuzzy, Inference rules, defuzzyfication methods of defuzzification.

Fuzzy Non-Linear Simulation 6 hours
Fuzzy Relational Equations, Partitioning, Non-Linear Simulation using fuzzy rule/knowledge based systems, Fuzzy Associative Memories (FAMs)

Fuzzy controller: 6 hours
Control strategies, general PID controller, Implementation of fuzzy systems in control, Direct fuzzy controller, Fuzzy P, PI and PID controller, Indirect fuzzy controller – fuzzy in handling the inner loops of control systems.

Nonlinear systems and adaptive fuzzy controller 4 hours
Nonlinear systems, modification in fuzzy systems for nonlinear control, Adaptive control, Adaptive control using fuzzy, fuzzy sliding mode controls.

Hybrid systems: 4 hours
Neuro- fuzzy and fuzzy genetic systems, applications to engineering problems.

Text Books:
Computer System Performance Analysis

Subject Code: 13MCA555        I.A. Marks: 50
Hours/Week: 4         Exam Hours: 3
Total Hours: 52         Exam Marks: 100

Introduction 8 Hours

Work Loads, Work Load Selection and Characterization 10 Hours
Types of Workloads, addition instructions, Instruction mixes, Kernels, Synthetic programs, Application benchmarks, Popular benchmarks, Work load Selection, Services exercised, level of detail, Representativeness, Timeliness, Other considerations in workload selection. Work load characterization Techniques; Terminology; Averaging, Specifying dispersion, Single Parameter histograms, Multi Parameter Histograms, Principle Component Analysis, Markov Models, Clustering.

Monitors, Program Execution Monitors and Accounting Logs 6 Hours
Monitors: Technology and classification; Software and hardware monitors, Software versus hardware monitors, Firmware and hybrid monitors, Distributed System Monitors, Program Execution Monitors and Accounting Logs, Program Execution Monitors, Techniques for Improving Program Performance, Accounting Logs, Analysis and Interpretation of Accounting log data, Using Accounting logs to answer commonly asked questions.

Capacity Planning and Bench Marking 5 Hours
Steps in Capacity Planning and Management: Problems in Capacity Planning, Common Mistakes in Bench Marking, Bench Marking Games, Load Drivers, Remote-Terminal Emulation; Components of an RTE; Limitation of RTEs

The Art of Data Presentation 4 Hours
Types of Variables, Guidelines for preparing Good Graphic charts, common Mistakes in preparing Charts, Pitorial Games, Gantt Charts, Kiviat Graphs, Schumacher Charts, Decision Maker’s Games

Experimental Design and Analysis 7 Hours

Queuing Models 12 Hours
Introduction: Queuing Notation: Rules for all Queues, Little’s Law, Types of Stochastic Process, Queuing Networks: Open and Closed Queuing Networks, Product form networks, queuing network models of computer systems, Operational Laws: Utilization Law, Forced Flow Law, Little’s Law; General Response Time Law; Interactive Response, Time Law; Bottleneck Analysis, Mean Value Analysis and Related Techniques, Analysis of Open Queuing Networks, Mean Value Analysis; Approximate MVA; Balanced Job Bounds; Convolution Algorithm, Distribution of jobs in a System, Convolution Algorithm for Computing G(N), computing performances using G(N), Timesharing Systems
Text Book:
   (Chapter 1,2,3,4,5,6,7,8,9,10,16,17,23,32,33,34,35)

Reference Books

Building Enterprise Applications

Subject Code: 13MCA556
IA Marks: 50

Hours/Week: 4
Exam Hours: 3

Total Hours: 52
Exam Marks: 100

Introduction
6 Hours
Enterprise applications and their types, software engineering methodologies, life cycle of raising an Enterprise application, introduction to skills required to build an Enterprise application, key determinants of successful Enterprise applications, and measuring the success of Enterprise applications

Inception of Enterprise Applications
8 Hours
Enterprise analysis, business modeling, requirements elicitation, use case modeling, prototyping, non-functional requirements, requirements validation, planning and estimation, *Case Study.

Architecting and Designing Enterprise Applications –part 1
8 Hours
Architecture, views and viewpoints, enterprise architecture, logical architecture, technical architecture and design, different technical layers, best practices, *Case Study.

Architecting and Designing Enterprise Applications –Part 2
10 Hours
Data architecture and design – relational, XML, and other structured data representations, Infrastructure architecture and design elements - Networking, Internetworking, and Communication Protocols, IT Hardware and Software, Middleware, Policies for Infrastructure Management, Deployment Strategy, Documentation of application architecture and design, *Case Study.

Construction Enterprise Applications
10 Hours
Defining a construction plan, defining a package structure, setting up a configuration management plan, setting up a development environment, introduction to the concept of Software Construction Maps, construction of technical solutions layers, methodologies of code review, static code analysis, build and testing, dynamic code analysis – code profiling and code coverage, *Case Study.
Testing and Rolling Out Enterprise Applications

10 Hours

Types and methods of testing an enterprise application, testing levels and approaches, testing environments, integration testing, performance testing, penetration testing, usability testing, globalization testing and interface testing, user acceptance testing, rolling out an enterprise application, *Case Study.

*Case Study Guidelines:-

Students should work in one case study (for Ex: - Telecom order Management System) to implement (and learn to use the tools to accomplish this task) the following (illustrative only)

- Understand a given business scenario and document the use case diagrams for the given scenario
- Identify the non-functional requirements for the given scenario and document it in the given template
- Create a logical architecture for the given business scenario documented in use case diagrams
- Create a data architecture for the given logical architecture
- Create a subset of design for the given logical architecture
- Create test cases (subset) as per the given template
- Code analysis of the given code base (case study)
- Testing the application of the given code base (case study) – Performance and Penetration testing.

By the end of the semester each student should submit one in detailed case study report for 10 Internal Assessment Marks.

Text Books

1. Raising Enterprise Applications – Published by John Wiley, authored by Anubhav Pradhan, Satheesha B. Nanjappa, Senthil K. Nallasamy, Veerakumar Esakimuthu
2. Building Java Enterprise Applications – Published by O'Reilly Media, authored by Brett McLaughlin

Reference Books

1. Software Requirements: Styles & Techniques – published by Addison-Wesley Professional
NET Laboratory
Subject Code: 13MCA57 I.A. Marks: 50
Hours/Week: 3 Exam Hours: 3
Total Hours: 42 Exam Marks: 50

PART – A
1. Write a Program in C# to demonstrate Command line arguments processing.
2. Write a Program in C# to demonstrate boxing and Unboxing.
3. Write a program to demonstrate Operator overloading.
4. Find the sum of all the elements present in a jagged array of 3 inner arrays.
5. Using Try, Catch and Finally blocks write a program in C# to demonstrate error handling.
6. Demonstrate Use of Virtual and override key words in C# with a simple program.
7. Write a program to demonstrate delegates.
8. Write a program to demonstrate abstract class and abstract methods in C#.
9. Write a program to illustrate the use of different properties in C#.
10. Demonstrate arrays of interface types (for runtime polymorphism) with a C# program.

PART – B
1. Consider the Database STUDENT consisting of following tables: tbl_Course (CourseID: int, CourseName: string) tbl_Student (USN: string, StudName: string, Address: string, CourseID: int, YrOfAdmsn: int)
Develop suitable windows application using C#.NET having following options:
1. Entering new course details.
2. Entering new student details.
3. Display the details of students (in a Grid) who belong to a particular course.
4. Display the details the students who have taken admission in a particular year.
2. Consider the Database BLOODBANK consisting of following tables: 
tbl_BloodGroup (BloodID: int, BloodGroup: string) 
tbl_Donor (DonorID: int, DonorName: string, Address: string, 
ContactNo: int, DOB: date, Gender: string, Weight: int, BloodID: int)
Develop suitable windows application using C#.NET having following options:
1. Entering Blood group details.
2. Entering new donor details.
3. Display the details of donors (in a Grid) having particular blood group.
4. Display the details of donors (in a Grid) based on gender.
5. Display the details of donors (in a Grid) based on age (above 18), weight (above 45KG) and 
   Gender(user’s choice).

3. Consider the Database STUDENT consisting of following tables: 
tbl_Course (CourseID: int, 
CourseName: string) 
tbl_Book (BookID: int, BookTitle: string, Author: string, CourseID: int) 
tbl_Student (USN: string, StudName: string, CourseID: int) 
tbl_BookIssue(USN: string, BookID: int, IssueDate: Date)
Develop suitable windows application using C#.NET having following options:
1. New Course Entry.
2. New Book Entry
3. New Student Entry
4. Issue of books to a student.
5. Generate report (display in a grid) showing all the books belonging to particular course.
6. Generate report (display in a grid) showing all the books issued on a particular date.
7. Generate report (display in a grid) showing all the books issued to a particular student.

4. Develop a Web Application using C#.NET and ASP.NET for an educational institution. The 
master page should consist of Institution Name, Logo and Address. Also, it should provide 
hyperlinks to Departments, Facilities Available and Feedback. Each department page and 
facilities page should be designed as static pages. The hyperlinks should navigate to these static 
pages in the form of Content Pages associated with Master Page designed. The Feedback page 
should have fields to enter Name, Email and Message with Submit and Cancel Buttons. Database 
should be created to store these three data.

5. Develop a Web Application using C#.NET and ASP.NET for a Bank. The BANK Database 
should consist of following tables: 
tbl_Bank (BankID: int, BankName: string) 
tbl_Branch (BranchID: int, BankID: int, BranchName: string) 
tbl_Account (AccountNo: int, 
BankID: int, BranchID: int, CustomerName: string, Address: string, ContactNo: int, Balance: 
real) (Note: AccountNo and BankID together is a composite primary key).
The master page of this web application should contain hyperlinks to New Bank Entry, New 
Branch Entry (of selected Bank), New Customer Entry (based on branch and bank) and Report 
Generation. The hyperlinks should navigate to respective content pages. These content pages 
provide the fields for respective data entry. The reports should be generated (display in grid) as 
below:
1. Display all records of particular bank.
2. Display all records of a branch of particular bank.
3. The balance should be displayed for the entered account number (Bank and Branch are 
   input through ComboBox controls and Account number is input through TextBox).

Note:
1. Students are required to execute one question from Part A and one from Part B.
2. Part A has to be evaluated for 20 marks and Part B has to be evaluated for 30 marks